Manufacturing Sentiment: Forecasting Industrial Production with Text Analysis*

Tomaz Cajner Leland D. Crane Christopher Kurz

Norman Morin Paul E. Soto Betsy Vrankovich

Preliminary and Incomplete—Please do not cite or distribute

March 2023

Abstract

This paper examines the link between industrial production and the sentiment expressed in text survey responses from U.S. manufacturing firms. We compare several natural language processing (NLP) techniques for classifying sentiment on our manufacturing-specific corpus, ranging from dictionary-based to deep learning methods. We find that deep learning models—partially trained on our data—achieve the highest sentiment classification performance on a manually-labeled sample. We assess the extent to which each sentiment measure, aggregated to monthly time series, can forecast industrial production. Our results suggest that the text responses provide information beyond the available numerical data and, improve out-of-sample forecasting. We also assess what drives the predictions made by the deep leaning models, and find that a relatively small number of words—associated with very positive/negative sentiment—account for much of the variation in the aggregate sentiment index.

JEL codes: C1, E17, O14

Keywords: Industrial Production, Natural Language Processing, Machine Learning, Forecasting

*This version is from March 2023. All authors are at the Federal Reserve Board of Governors. We thank the Institute for Supply Management for access to and help with the manufacturing survey data that underlie the work described by this paper. The analysis and conclusions set forth here are those of the authors and do not indicate concurrence by other members of the research staff or the Board of Governors.

1 Introduction

In recent years there has been an explosion of interest in natural language processing (NLP) within finance and macroeconomics. The use of text data to forecast and assist in model estimation is becoming increasingly commonplace. Still, there are many open questions around the use of NLP in macroeconomics. For example, which of the numerous available methods work best, and work best in specific contexts? Are off-the-shelf tools appropriate, or are there greater returns to specializing models to the data in hand? How useful is text for forecasting real output indicators, such as manufacturing output? This paper addresses these questions, using a novel dataset and a variety of NLP methods.

Our primary data source is the monthly survey microdata for the Institute of Supply Management's (ISM) Report on Business. The survey is taken by purchasing managers at a representative sample of U.S. manufacturing firms. Part of the survey consists of categorical-response questions about aspects of their current operations, including production levels, inventories, backlogs, employment, and new orders. The answers to these questions are aggregated into the widely-reported ISM diffusion indexes. But the survey also includes free-form text boxes, where purchasing managers can provide further comments either in general or about specific aspects of their businesses; these free-form comments are a novel potential source of signal and our focus in this paper.

The first step of our analysis is to evaluate various natural language processing (NLP) techniques for classifying sentiment in our data. Our context is fairly specific: the data are manufacturing-sector purchasing managers opining about their business outlook, without much discussion of financial conditions. While there are numerous sentiment classification tools and techniques available, many were developed for vastly different contexts, such as social media posts (Nielsen, 2011). Even within economics and finance, most work has focused on the financial side (Araci, 2019; Correa et al., 2021; Huang et al., 2022). The lack of results for related datasets motivates our assessment of a wide variety of NLP techniques. One common approach is to count the frequency of words within a sentiment dictionary. Economists initially used positive and negative words from psychology literature, but have since moved on to using domain-specific words (e.g., Correa et al. (2021)) and using simple word counts to measure uncertainty (see Baker et al. (2016) and Gentzkow et al. (2019)). While this method is simple, it may fail to capture negation, synonyms, and often requires context-specific dictionaries that may not be available. More recent-developed techniques employ deep learning methods that account for the nuances of language. These models are *pre-trained*: the parameters are set by exposing the model to a large corpus of text, such as

the entirety of Wikipedia. The pre-trained models can be used to classify sentiment directly, or they can be further trained ("fine-tuned") on a specific dataset. The latter approach attempts to get the best of both worlds: a solid ability to parse language from the exposure to extremely large training data, plus the context-specific nuance from the fine-tuning data.

Our first contribution is to compare the accuracy of these different methods, using a sample of hand-coded comments from our dataset. The range of available NLP tools and methods is vast, and there have been very few comparisons of NLP methods for economics and finance-related tasks (Kalamara et al. (2022) is a notable exception.) Deep learning gets enormous attention, but it is ex-ante unclear whether it should outperform carefully curated dictionaries in our context. We find that deep learning does have an advantage on our data, in part because the brevity of the comments means that many comments have no overlap with dictionary terms. In addition, we find that there is value in specializing the models to our data: the fine-tuned models have the highest sentiment classification accuracy on a hold-out sample. These results point to the advantages of using pre-trained models, as well as carfully specializing them to the task at hand. Our hope is that these results help guide other economists when deciding between NLP approaches.

Next, we investigate the relationship between sentiment expressed by purchasing managers and future manufacturing output, as measured by the manufacturing component of industrial production. Our baseline model includes–among other controls—some of the ISM diffusion indexes, so the test is whether the sentiment indexes have additional information beyond the ISM categorical responses data. We find that most dictionary-based text variables are not predictive of future manufacturing output, with the exception of a curated financial stability-specific dictionary. On the other hand, sentiment variables generated using deep learning are predictive of future manufacturing output. Through out-of-sample exercises, we also find that the financial stability dictionary and deep learning techniques significantly reduce the mean squared errors. Overall, our results suggest that purchasing managers' survey responses contain useful forward-looking information, and that sentiment-based measures can improve the accuracy of forecasts of manufacturing output.

Finally, we investigate what features of the data are most important to our fine-tuned deep learning model. To shed light on these black box models, we use a standard method—Shapley decompositions—to score the contribution of each individual word. Our results point to a sensible interpretation of our deep learning models: the most positive words include "brisk", "excellent", "booming", "improve", and "efficient"; among the most negative words are "unstable", "insufficient", "fragile", "inconsistent", and "questionable". In addition, we

find that average sentiment and changes in average sentiment are largely accounted for by the words with the most extreme (positive or negative) sentiment scores, with the vast majority of words playing little role.

Our paper contributes to two strands of literature. First, our comparison of NLP techniques for measuring sentiment adds to the growing body of literature incorporating NLP into economic and financial research. Since the seminal work of Tetlock (2007), several studies have used dictionary-based methods (Baker et al., 2016; Hassan et al., 2019; Young et al., 2021). Refined lexicons for specific contexts have been shown to improve performance in measurement and forecasting (Correa et al., 2021; Gardner et al., 2022; Sharpe et al., 2017). Machine learning techniques have also been used to select word lists (Manela and Moreira, 2017; Soto, 2021). More recent papers incorporate more sophisticated machine learning methods to extract the tense and topic of texts (Angelico et al., 2022; Hanley and Hoberg, 2019; Hansen et al., 2018; Kalamara et al., 2022). Advances in NLP, particularly the use of deep learning techniques, have significantly improved sentiment classification (Heston and Sinha, 2017; Araci, 2019; Huang et al., 2022).

Second, we contribute to the literature on predicting industrial production (D'Agostino and Schnatz, 2012; Lahiri and Monokroussos, 2013; Ardia et al., 2019; Cimadomo et al., 2022; Andreou et al., 2017). Our analysis of the relationship between sentiment and industrial production provides new insights into the role of unstructured text data in economic forecasting. By comparing various NLP techniques, we are able to identify which methods are most effective for classifying sentiment and incorporating them into predictive models of industrial production.

The remainder of the paper is structured as follows. Section 2 presents our data. Section 3 reviews how we measure sentiment from the textual survey data, and Section 4 reviews the empirical strategy and results. Section 5 discusses the mechanism with which firm survey responses predict industrial production. Section 6 concludes.

2 Data

The primary data for this study comes from the Institute of Supply Management (ISM). Each month, the ISM conducts a survey of purchasing managers from a sample of manufacturing firms in the United States. The ISM data is timely and relevant. Indeed, as highlighted in Bok et al. (2018), not only does such survey data provide important signal about the state of the economy, but the ISM data in particular provides the "earliest available information

for the national economy on any given quarter."

The ISM survey includes a series of questions about the respondents' operations, including their production levels, new orders, backlog, employment, supplier delivery times, input inventories, exports, and imports. These questions have a categorical response, allowing the purchasing managers to specify whether these metrics have increased, decreased, or stayed the same between last month and the current month. These categorical responses are aggregated into publicly-released diffusion indexes, discussed more below. All the survey questions are shown in Table 1. In addition to the categorical response, purchasing managers can provide further explanation in accompanying text boxes (the questions with "Free text" response in the table.) There are free text response questions accompanying nearly every categorical question, asking for the reason for the response.[1] In addition there is a "Remarks" field at the beginning, where the respondent can put any general remarks they wish. The text responses are briefly excerpted in the ISM's data release to provide context for the diffusion indexes, but otherwise are not released publicly.

The survey dates back to the 1930s, and our dataset covers the roughly 30,000 firm-month observations from 2007 to 2019. Figure 1 shows the number of firms in the sample over time. The sample size has varied, increasing to nearly 300 respondents by the end of 2019. The figure also shows that the majority of respondents provided clarifying remarks in their responses, with word counts ranging from 10 to 29 words on average per month. We observe a sudden increase in word count around 2018, which appears to be due to the tariffs imposed on China. After removing responses that contain the word "tariff," we observe a smoother increase in word counts (see Figure A1 in the appendix for further details).

Table 2 provides a summary of the text responses. Nearly 49 percent of the general remarks sections contain text, while the next most common sections containing text are those related to production, new orders, and input inventories. The last row shows statistics for all the text fields concatenated together: 68 percent of firm-month observations have any text at all, and the text is about 16 words long on average. The average word count is highest for the General Remarks section, with an average of 8 words used in these responses. When considering only those responses that contain text, the average word count for the General Remarks section increases to 17 words.

Turning from the ISM survey microdata, we use several time series in our forecasting exercises. Our focus is on forecasting the manufacturing industrial production (IP) index. We use real time data on the right hand side, reflecting what policy makers knew at the time,

[1]The exception is the general "Remarks" question, which only asks for a free form response.

and forecast the fully revised series. In addition to IP series, we use the ISM diffusion indexes as regressors. The diffusion indexes are aggregations of the categorical response questions in the survey. For example, the production diffusion index is a weighted average of the responses to the production question (paraphrasing, "Is production higher/the same/lower than last month?"), with the "Higher" responses getting weight 100, "Same" responses getting weight 50, and "Lower" responses getting weight 0. The formula for the diffusion index in period t, with N_t total firms responding is shown in equation (1):

$$D_t = \frac{1}{N_t} \sum_{i=1}^{N} [100 \cdot \mathbf{1}\{\text{Response } i \text{ is "Higher"}\} + 50 \cdot \mathbf{1}\{\text{Response } i \text{ is "Same"}\}] \quad (1)$$

These diffusion indices have values between 0 and 100, with 0 indicating that all respondents say things are worse and 100 indicating that all respondents say things are better. The ISM publishes indexes for each question, as well as a "PMI Composite", which is an equally-weighted average of the diffusion indexes for new orders, production, employment, supplier deliveries, and inventories.

3 Measuring Sentiment

Our goal is to extract useful information from the ISM survey text responses. We focus on sentiment analysis: measuring the positive or negative sentiment expressed by the purchasing managers. Even focusing on sentiment analysis, the wide range of natural language processing (NLP) techniques available can make it challenging to choose an appropriate method. In this section we discuss the methods we use, leaving a complete description of the approaches to the Appendix 7.

One of the simplest methods for measuring sentiment is dictionary-based analysis, which involves counting the frequency of a predetermined list of sentiment words in the text. We use common sentiment dictionaries from the psychology literature, such as the Harvard (Tetlock, 2007) and AFINN (Nielsen, 2011) word lists. However, we also recognize that certain words that may be considered negative in other contexts may not be considered negative in the context of finance, such as "taxing" or "liability". As such, we also apply finance-specific word lists, including the sentiment word list from Loughran and McDonald (2011) (henceforth, "LM") and the financial stability word list from Correa et al. (2021). For all dictionaries, we score comments on a scale of -1 to +1, using the percent of total words in the comment that are positive less the percent of total words that are negative. When

we require discrete classifications, as in Figure 2, we classify the comment as positive if the score is greater than zero, negative if it is less than zero, and neutral if it equals zero.

Another approach to sentiment analysis involves fitting a model to the data. We try several variations on this theme, ranging from regressions on word counts to customized neural networks. Unlike the dictionary methods, all of these approaches require labeled data: a sample of observations that have already been classified, which is used to fit the model and classify the remaining observations.

3.1 Human-labeled data

We create a labeled dataset from a randomly selected subsample of 2,000 of the text responses to the individual questions.[2] Each response was classified for sentiment by two economists using the following question as a guide: "*Is this comment consistent with manufacturing IP rising month over month?*" The classifications were either positive, neutral, or negative, where "neutral" includes cases where is it is impossible to determine the sentiment. Both economists agreed on the sentiment classification for roughly 1,500 cases. This subsample is further split into a "training" dataset, used to fit the models, and "test" dataset, used to assess the relative merits of the models.[3]

3.2 LASSO

The first model-based approach is a logistic LASSO regression. This treats the data as a "bag of words", where all that matters is the counts of individual words, not their order. In the regressor matrix each unique word is placed in a column, with the counts of that word populating the column. This leads to a large number of regressors (the number of unique words in the training data), motivating the use of LASSO to regularize the regression. The dependent variable is the categorical variable $y_i \in \{1, 2, 3\}$, representing a negative, neutral or positive classification, respectively.

3.3 Deep Learning Models

Deep learning models have gained popularity in recent years, driven by their impressive performance on language-related tasks. Much of the progress has occurred within a particular

[2]Note, that the categorical responses can be considered a kind of label. In Section XXXXX we investigate how well models can predict the categorical response from the associated text.

[3]The test data consists of 10 percent of the human-labeled dataset, and is not used by any of the models during training.

class of deep learning models called *transformers* (see, e.g., Devlin et al. (2018), Radford et al. (2018), Ouyang et al. (2022), Chung et al. (2022), and Touvron et al. (2023)). The defining feature of transformers—relative to other neural network architectures—is a mechanism called *attention*; a way to interact words within a sentence, allowing the context of a particular word to influence the meaning. A full explanation of transformers and the attention mechanism is beyond the scope of this paper, but we do provide a brief summary in the Appendix. The important points are that (unlike dictionaries and bag-of-words approaches), transformers take into account interactions between words, word order, and context-dependent meanings (polysemy).

One notable transformer model is "BERT", or Bidirectional Encoder Representations from Transformers, developed by Devlin et al. (2018). It is important to note that BERT is a pre-trained model: Devlin et al. (2018) specified the architecture and then trained the model on a corpus including the entirety of (English) Wikipedia and a number of books. The model is large by the standards of the economics literature, with roughly 110 million parameters. We use several versions of BERT in this paper.

By default, the off-the-shelf BERT model produces sentence embeddings: Given a sentence-length piece of text, it returns a 768-dimensional vector representing the sentence. Intuitively, sentences with similar meaning ought to have embedding vectors close to each other. BERT can be used as a classifier by adding an additional layer on top of it, essentially a logistic regression that takes the embedding vector as the input and returns class probabilities. Note that this requires some labeled data to fit the logit.

BERTs open access saves researchers the expensive cost of training a large language model, while still allowing them to leverage the pre-trained version for their specific needs, in a concept known as 'fine-tuning.' In the financial domain, specialized BERT models have been developed to account for the unique characteristics of financial and economic text. Two prominent examples are Huang et al. (2022) (which we refer to as FinBERTv1) and Araci (2019) (which we refer to as FinBERTv2.) FinBERTv1 uses the BERT architecture but is trained from scratch on SEC filings, equity reports, and earnings conference call transcripts. The sentiment classification layer is trained on the human labeled AnalystTone dataset Huang et al. (2014).[4] FinBERTv2 was initialized with the pretrained BERT weights and further pre-trained on a corpus of Reuters news articles, which tend to focus on financial news. The sentiment classification layer was trained on the human-labeled Financial

[4]Specifically, the model is `yiyanghkust/finbert-tone` from the Huggingface model hub, a classification fine-tuned version of "FinBERT-FinVocab uncased" in Huang et al. (2022).

PhraseBank dataset from Malo et al. (2014).[5]

While FinBERTv1 and FinBERTv2 can do a good job parsing financial news and regulatory filings, our data are more focused on topics like order backlogs, production difficulties, inventories, and delivery times, which are not commonly found in financial corpora. After reviewing the text responses from the ISM survey, we found examples suggesting that FinBERTv1 and FinBERTv2 have some difficulty with the language. For example, the comment "slight up-tick inventory to account for slight up-tick in production" is coded as positive by the economists: it implies increased production, and an increase in input inventories to support that higher level of production. But this passage is classified as neutral by FinBERTv1 and negative for FinBERTv2. These issues motivate our use of the human-labeled dataset to fine-tune or train from scratch our own models. First, we estimate our own transformer model using the training dataset and a relatively small number of parameters. We call this model, *TF-Small* (TF for 'transformer')[6] Second, we fine-tune BERT with our manually labeled training examples, and call the resulting model *Fine-Tuned BERT*. Fine-Tuned BERT benefits from the large size and extensive training of the base BERT model, but is explicitly tuned on the language relevant for our task. As we shall see below, this results in good performance.

When applying the transformer-based models, we use the predicted most likely class (positive, neutral or negative) as the output and code these as +1, 0 and -1 respectively. In Section 5 we exploit the predicted class probabilities as well.

Overall, we propose eight models for sentiment classification. The four dictionary-based methods are the Harvard, AFINN, Loughran and McDonald (2011), and financial stability (Correa et al. (2021)) dictionaries, and the four transformer models are FinBERTv1, FinBERTv2, *TF-Small*, and Fine-Tuned BERT.

3.4 Comment-Level Classification Results

We evaluate the accuracy of each model on the test human-labeled dataset, as shown in Figure 2.[7] The confusion matrix for each model tabulates the percent of observations with a given human "true" classification (which varies across rows) and the model-based predicted

[5]This model is `ProsusAI/finbert` on the Huggingface model hub.

[6]We use the *Keras* library to build a simple encoder-only transformer model with input embedding dimension of 16 and an output sentiment layer with similar dimensions.

[7]While the test dataset contains 154 observations, we report predictions for only the 111 observations for which a categorical response is provided, excluding the General Remarks responses. This step was taken so as to make the evaluation sample for the categorical response similar to the evaluation sample for the other models.

classification (which varies across columns.) Overall accuracy is reported at top of each matrix. We begin by considering whether the categorical response for each comment is predictive of the human label applied to the corresponding text. For example, if the human label for a new orders response is positive, we'd like to know how often was the categorical response that new orders are higher than last month. We find an overall accuracy of 66.7%, suggesting that the sentiment in the text responses is not fully redundant with the categorical responses. Interestingly, of the nearly 20% of responses that have a human-labelled neutral sentiment, 70% are associated with a positive categorical response variable.

The Harvard, AFINN, Loughran-McDonald, and Stability dictionaries all have accuracy scores around 30%. The low accuracy is due to the fact that nearly all responses are predicted to be neutral. Dictionary-based methods can only produce a positive or negative classification if either positive or negative words appear in the text, and the short comments in our data often do not contain any of the words in the dictionaries. Due to this limitation, the dictionary-based methods are not accurate at classifying the comments.

FinBERTv1 and FinBERTv2 perform better, with accuracies of 76.6% and 70.3%, respectively. Both of these models are better able to classify actual neutral responses, but both tend to over-predict neutral classifications. The best performing models are TF-Small and Fine-Tuned BERT models, with accuracies of 84.7% and 85.6% respectively. It appears that the good performance of these models is largely due to having seen examples of manufacturing-specific text, as well as survey-specific examples of positive, negative, and neutral responses.

We next run the eight sentiment classifiers on all available observations, and average the sentiment scores by month; these monthly averages are what will feed into the forecasting models in Section 4.[8] Table 3 collects the summary statistics for the monthly series. The dictionary-based monthly averages tend to have a mean close to zero and a small standard deviation, as a result of the infrequent usage of words appearing in the dictionaries. In contrast, the transformers models have larger (in absolute value) means and standard deviations, as each firm-month response is assigned a value of 1 for positive, 0 for neutral, and -1 for negative, rather than a percentage of sentiment words in the sample. Across the entire sample, the average sentiment of most transformer models is negative. This finding aligns with previous research on text analysis in finance suggesting a negative bias in textual data (Tetlock (2007)).

[8]Note that the comment-level sentiment scores all range between -1 and +1.

4 Time Series Results

Our forecasting exercises focus on predicting monthly manufacturing output growth, as measured by the Federal Reserve Board's Industrial Production statistics. The real time data flow is important to understand, and is as follows:

- The ISM data for a month t are typically released on the third business day of month $t+1$
- The first IP data for month t are typically released around the 15th of month $t+1$
- The IP estimates for a month t are revised several time over the subsequent months and years, as more product data becomes available and benchmark revisions are incorporated. The monthly revisions all take place as part of the subsequent month's IP releases, so the first monthly *revision* to IP for month t is released around the 15th of month $t+2$, the second revision occurs around the 15th of $t+3$, etc.

Our baseline forecasting model is as follows:

$$\Delta IP_t^{current} = \alpha + \beta_1 \Delta IP_{t-1}^{t^*} + \beta_2 \Delta IP_{t-2}^{t^*} + \beta_3 \Delta IP_{t-3}^{t^*} + \delta x_t^{t^*} + \epsilon_t \tag{2}$$

where $\Delta IP_t^{current}$ is the fully revised, current-vintage growth rate of manufacturing output in month t. The superscript t^* denotes a variable as reported on the eve of the month t G.17 IP data release: the real-time vintage relevant for forecasting ΔIP_t just prior to its first print. Thus $\Delta IP_{t-1}^{t^*}$ is the estimate of month $t-1$ from the initial month $t-1$ data release (released around the middle of month t), and $\Delta IP_{t-2}^{t^*}$ is the *revised* estimate of month $t-2$ from the month $t-1$ data release (again, released around the middle of month t). The vector $x_t^{t^*}$ collects the ISM metrics for month t. These are available well before the month t IP data, and so may be particularly useful for forecasting. For the baseline model x_t contains only the the composite PMI index, an average of five of the ISM diffusion indexes.

Table 4 presents the in-sample estimation of our baseline model with the added text measures. In column (1), we see that the baseline model has an R-squared of 28.8% with a positive and statistically significant relationship between the composite PMI and IP growth. The following columns show that the LM, Harvard, and AFINN dictionaries are not statistically significant, and only lead to small improvements in R squared. Column (5) shows a positive and significant effect (at the 10% level) of sentiment as measured with the Stability

dictionary, and a larger increase in R squared. Moving to columns 6-9, all four transformer-based sentiment measures are positively and significantly related to manufacturing growth. The deep learning sentiment measures also have improved explanatory power, seen by the two to three percentage point increase in R-squared.

Next, we assess the out-of-sample performance of the sentiment indexes. Table 5 shows the Diebold-Mariano test statistic comparing the forecast of the baseline model with each text-augmented baseline model over the period 2018m1-2019m12. Each cell displays the out-of-sample RMSE and DM test statistic. In the top row—for our preferred specification—we see that the LM dictionary-based text measure increases the RMSE, while the Harvard and AFINN dictionary-based text measures reduce the RMSE slightly. The Stability dictionary results in an RMSE reduction of nearly 11%. Similarly, the transformer-based sentiment measures significantly reduce the out-of-sample forecast errors, both statistically and economically. The other rows in the table show alternative specifications: only including the PMI index as a control, only using lagged manufacturing growth as a control, replacing the PMI composite with new orders, and including several ISM diffusion indexes as controls. In nearly all cases, the Stability dictionary the and transformer-based models significantly reduce the out-of-sample RMSE.

5 Interpretation

The results in Section 4 suggest that the sentiment indexes, and fine-tuned BERT in particular, provide additional forecasting power. However, BERT is very much a black box, and it is far from obvious what drives its behavior. In this section we provide supporting evidence to help interpret the BERT results. It is difficult to explain or interpret the predictions from deep learning models like transformers. It is unclear which features of the model architecture or the data lead to certain predictions being made. Fortunately, there is an active field of research in interpretable machine learning, and many methods have been proposed to deal with these issues. We will use one such method—Shapley decompositions—as the basis of our work.

5.1 Shapley Decompositions

Shapley decompositions are used in machine learning to deal with the nonlinear relationships between the dependent variable and independent variables (Lundberg and Lee (2017)), drawing on cooperative game theory results from Shapley (1953). Given (1) an observation, and

(2) the prediction of the model, the Shapley decomposition estimates the additive contribution of each feature to the prediction. Each contribution is relative to a "null value" for the feature. For numeric data the null value might be the mean. Roughly speaking, the Shapley decomposition calculates the marginal contribution of switching a given feature from its null value to the observed value, averaging across all possible null/observed permutations for the *other* features. The averaging across permutations ensures that the resulting contributions have good properties, including additivity. The contributions to the prediction add up to the prediction exactly.

In our context, an observation is a single ISM comment, and the features are the individual words. BERT provides three predictions for each observation: the probability of being in the negative, neutral and positive classes. Rather than deal with this vector, we calculate the net positive score: Pr[positive class] − Pr[negative class], and use this as the prediction. The net positive score is analoguous to the diffusion index formula, and reduces the model output to a single number between -1 and +1.

To understand how the Shapley decomposition operates in our context, consider the example comment "`Business continues to be slow`". Fine-tuned BERT predicts this comment is positive with probability 0.078, with a net positive probability of -0.76. The Shapley decomposition proceeds by replacing subsets of the words with a special token, `[MASK]`.[9] BERT interprets `[MASK]` as meaning that there is a real, unknown word in that place in the comment. BERT continues to make predictions for the class of the comment even when words are masked; these predictions are based on the remaining unmasked words and the positions of the words in the comment.

The marginal contribution of the word "`slow`" can be calculated as the difference between the net positive probability of "`Business continues to be slow`" and "`Business continues to be [MASK]`". However another plausible estimate of the marginal contribution would be, e.g., the difference between "`[MASK] continues to be slow`" and "`[MASK] continues to be [MASK]`". The Shapley decomposition iterates over the various masking permutations to arrive at an average marginal contribution.[10]

It is worth noting here that the Shapley decomposition is not a structural explanation, nor does it imply any casual relationship. It is an accounting identity that can be imposed on any model. For our purposes, it is useful for linearizing the the relationship between

[9] In NLP "tokens" are the basic unit of observation, roughly speaking they can be words or word parts.

[10] In practice, calculating every permutation requires 2^N model evaluations for a sequence with N tokens, which can become very costly even for comments around 16 words long. The SHAP package for Python circumvents this issue by sampling.

tokens and the aggregate sentiment index.

After running the Shapley decomposition on all the comments, we have *Shapley scores* for each token in each comment. The Shapley scores for the tokens in a given comment add up to the net positive probability for that comment. The contribution of a token can vary across comments, because BERT's predictions are not a linear function of the tokens. This is part of the advantage of BERT: tokens may have different meanings depending on the context. But we can assess the *average* contributions of the tokens to check that the results are sensible. Table 6 shows the words with the most positive and negative Shapley scores. The words in each group appear quite reasonable. We can also examine the distribution or words across scores: Figure 4 plots the density of words across Shapley scores. The density is winsorized at ± 0.05 to make the central mass visible. The weighted density, in black, shows the distribution weighted by the number of occurrences in the corpus. The vermillion unweighted density counts each unique token in the vocabulary equally. Note that many tokens have scores close to zero, particularly in the weighted plot. As the Shapley scores can range between positive and negative 1, it might be puzzling why so much mass is concentrated on the (-0.05,0.05) interval. Part of the reason is simply due to the length of the comments: if comments are on average 16 words long, a random word will—on average—only contribute 1/16th to the comment's score (which is bounded on (-1,1)). In addition, many of the tokens are filler words or word parts, e.g., the token "the" has a Shapley score of 0.003.

5.2 Time Series Properties

Our main interest is not so much in which words contribute the most to the classification of individual comments, but in the *change* in the *aggregate* sentiment time series. The Shapley decomposition can help here as well. To illustrate this, we focus on the change in sentiment between the first quarter of 2007 and the first quarter of 2009. These two dates are near the peak and trough of the business cycle, respectively, and picking the same quarter each year lets us ignore seasonality. The fine-tuned BERT sentiment index fell by about -0.4 between these two periods. We will determine what changes in language accounted for this decline.

First, it is useful to introduce and approximate a linear sentiment index. The actual sentiment index is a nonlinear function of the the probabilities returned by BERT: The sentiment indexes uses the comment *classifications* as -1 (negative), 0 (neutral) and 1 (positive), and the predicted class is nonlinear in the the underlying probabilities. It turns out that this diffusion index is well-approximated by the simple average of the comment probabilities. This simplification means that the (approximate) sentiment index is just an average of the

individual token contributions, aggregating across first tokens in a comment and then across comments. The contribution of each token to the aggregate is then just the sentiment of the token times the number of occurrences.

A basic question is whether the decline in sentiment around the Great Recession was due to (1) a change in the mix of tokens being used (i.e. a change in the pattern of occurrences) or (2) changes in the sentiment assigned to the tokens in the two periods. The results in Table XXXXX suggest that most of the effect is attributable to the first factor, the change in occurrences with sentiment fixed. To some extent this is reassuring: it would be harder to interpret a situation where the connotation of the words was driving aggregate sentiment.

The next question is which words contribute most to the decline in aggregate sentiment. Given the results above we can weight words by their full-sample average sentiment, rather than period-specific sentiment, and focus on the change in occurrences. Then we can calculate the contribution of each token to the aggregate sentiment score as the token's average sentiment score, multiplied by the number of occurrences, divided by the number of comments in that period. Note that if we sum this contribution across all tokens for a period, we get the sum of all the token Shapley scores, divided by the number of comments—the average *comment* score, in other words. Thus these contributions map directly from individual token occurrences to the aggregate (approximate) sentiment index.

To reduce the dimensionality of the problem we group the tokens into 20 quantiles, based on their average Shapley score. Summing the contributions within each quantile, we can see which quantiles are most important for the aggregate index, and which quantiles explain changes in the aggregate index. Figure 5 shows these comparisons. The black line shows the contributions by quantile for 2007:Q1. The bottom two or three quantiles have large, negative contributions to the aggregate index in that period. The top two quantiles have fairly large positive contributions, while all the others are negligible. We conclude that the aggregate score is product mostly of the extreme quantiles—where each token occurrence is highly weighted—rather than the intermediate quantiles. This is not obvious ex ante, we might have imagined that high token counts in the intermediate quantiles more than offset small weights.

The extreme quantiles also appear to account for the *change* in aggregate sentiment. Comparing the blue and the black lines, the more negative sentiment in 2009:Q1 is a result of more tokens being used in the bottom two quantiles, and somewhat fewer tokens being used in the top quantile, which the intermediate quantiles largely irrelevant. Tables 7 and 8 list the bottom- and top-quantile words.

To summarize, the aggregate sentiment index is a nonlinear function of the BERT model's predictions. Shapley values allow us to begin interpreting the contributions of individual tokens to the index, and it turns out that tokens have mostly invariant Shapley scores, with variation in token counts driving aggregate changes. Tokens with Shapley scores close to zero contribute very little to the aggregate score, even though they account for much of the distribution of tokens. Furthermore, the very top and bottom quantiles of the score distribution are by far the most important in accounting for changes in the index, with the Great Recession leading to both fewer very positive words and more very negative words.

6 Conclusion

In this study, we examine the relationship between manufacturing sentiment and industrial production growth, an important indicator for macroeconomic forecasting. To evaluate the effectiveness of the sentiment measures, we compare dictionary-based and deep learning methods to human labelled sentiment scores. Our results show that context-specific dictionary-based methods and deep learning techniques perform best in mimicking human sentiment. In addition, when estimating out-of-sample industrial production growth, we find that a finance-specific dictionary sentiment measure and all deep learning sentiment measures significantly improve forecasting accuracy.

Our comparison of different sentiment measures can assist future researchers in choosing the most appropriate methodology for their text analysis studies. Our findings suggest that deep learning techniques benefit from the use of manually labelled text data, and that context-specific dictionaries outperform general purpose dictionaries in out-of-sample exercises. Moreover, the improvements in industrial production forecasts achieved through the use of survey responses suggest that other macroeconomic variables may also benefit from the inclusion of unstructured data such as text.

References

Andreou, Elena, Patrick Gagliardini, Eric Ghysels, and Mirco Rubin, "Is Industrial Production Still the Dominant Factor for the US Economy?," 2017.

Angelico, Cristina, Juri Marcucci, Marcello Miccoli, and Filippo Quarta, "Can we measure inflation expectations using Twitter?," *Journal of Econometrics*, 2022, *228* (2), 259–277.

Araci, Dogu, "Finbert: Financial sentiment analysis with pre-trained language models," *arXiv preprint arXiv:1908.10063*, 2019.

Ardia, David, Keven Bluteau, and Kris Boudt, "Questioning the news about economic growth: Sparse forecasting using thousands of news-based sentiment values," *International Journal of Forecasting*, 2019, *35* (4), 1370–1386.

Baker, Scott R, Nicholas Bloom, and Steven J Davis, "Measuring economic policy uncertainty," *The Quarterly Journal of Economics*, 2016, *131* (4), 1593–1636.

Bok, Brandyn, Daniele Caratelli, Domenico Giannone, Argia M. Sbordone, and Andrea Tambalotti, "Macroeconomic Nowcasting and Forecasting with Big Data," *Annual Review of Economics*, 2018, *10* (1), 615–643.

Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei, "Scaling Instruction-Finetuned Language Models," 2022.

Cimadomo, Jacopo, Domenico Giannone, Michele Lenza, Francesca Monti, and Andrej Sokol, "Nowcasting with large Bayesian vector autoregressions," *Journal of Econometrics*, 2022, *231* (2), 500–519.

Correa, Ricardo, Keshav Garud, Juan M Londono, and Nathan Mislang, "Sentiment in central banks' financial stability reports," *Review of Finance*, 2021, *25* (1), 85–120.

D'Agostino, Antonello and Bernd Schnatz, "Survey-based nowcasting of US growth: a real-time forecast comparison over more than 40 years," 2012.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

Gardner, Ben, Chiara Scotti, and Clara Vega, "Words speak as loudly as actions: Central bank communication and the response of equity prices to macroeconomic announcements," *Journal of Econometrics*, 2022, *231* (2), 387–409.

Gentzkow, Matthew, Bryan Kelly, and Matt Taddy, "Text as data," *Journal of Economic Literature*, 2019, *57* (3), 535–74.

Hanley, Kathleen Weiss and Gerard Hoberg, "Dynamic interpretation of emerging risks in the financial sector," *The Review of Financial Studies*, 2019, *32* (12), 4543–4603.

Hansen, Stephen, Michael McMahon, and Andrea Prat, "Transparency and deliberation within the FOMC: a computational linguistics approach," *The Quarterly Journal of Economics*, 2018, *133* (2), 801–870.

Hassan, Tarek A, Stephan Hollander, Laurence Van Lent, and Ahmed Tahoun, "Firm-level political risk: Measurement and effects," *The Quarterly Journal of Economics*, 2019, *134* (4), 2135–2202.

Heston, Steven L and Nitish Ranjan Sinha, "News vs. sentiment: Predicting stock returns from news stories," *Financial Analysts Journal*, 2017, *73* (3), 67–83.

Huang, Allen H., Amy Y. Zang, and Rong Zheng, "Evidence on the Information Content of Text in Analyst Reports," *ERN: Econometric Modeling in Financial Economics (Topic)*, 2014.

Huang, Allen H, Hui Wang, and Yi Yang, "FinBERT: A large language model for extracting information from financial text," *Contemporary Accounting Research*, 2022.

Kalamara, Eleni, Arthur Turrell, Chris Redl, George Kapetanios, and Sujit Kapadia, "Making text count: economic forecasting using newspaper text," *Journal of Applied Econometrics*, 2022, *37* (5), 896–919.

Lahiri, Kajal and George Monokroussos, "Nowcasting US GDP: The role of ISM business surveys," *International Journal of Forecasting*, 2013, *29* (4), 644–658.

Loughran, Tim and Bill McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," *The Journal of finance*, 2011, *66* (1), 35–65.

Lundberg, Scott M. and Su-In Lee, "A Unified Approach to Interpreting Model Predictions," in "Proceedings of the 31st International Conference on Neural Information Processing Systems" NIPS'17 Curran Associates Inc. Red Hook, NY, USA 2017, p. 4768–4777.

Malo, Pekka, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," *Journal of the Association for Information Science and Technology*, 2014, *65* (4), 782–796.

Manela, Asaf and Alan Moreira, "News implied volatility and disaster concerns," *Journal of Financial Economics*, 2017, *123* (1), 137–162.

Nielsen, Finn Årup, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," *arXiv preprint arXiv:1103.2903*, 2011.

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe, "Training language models to follow instructions with human feedback," 2022.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, "Language Models are Unsupervised Multitask Learners," 2018.

Shapley, L. S., "A Value for n-Person Games," in Harold William Kuhn and Albert William Tucker, eds., *Contributions to the Theory of Games (AM-28), Volume II*, Princeton: Princeton University Press, 1953, pp. 307–318.

Sharpe, Steven A, Nitish Ranjan Sinha, and Christopher Hollrah, "What's the story? A new perspective on the value of economic forecasts," 2017.

Soto, Paul E, "Breaking the Word Bank: Measurement and Effects of Bank Level Uncertainty," *Journal of Financial Services Research*, 2021, *59* (1), 1–45.

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

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample, "LLaMA: Open and Efficient Foundation Language Models," 2023.

Young, Henry L, Anderson Monken, Flora Haberkorn, and Eva Van Leemput, "Effects of supply chain bottlenecks on prices using textual analysis," 2021.

Tables

Table 1: ISM Survey Instrument

Field	Type of Response
Remarks:	Free text
Indicate if the current month's level of production (units, not dollars) compared to the previous month is:	Higher/Same/Lower
Reason if higher or lower:	Free text
Indicate if the current month's level of new orders (i.e., new sales orders from customers) (units, not dollars) for finished products compared to the previous month is:	Higher/Same/Lower
Reason if higher or lower:	Free text
Indicate if the current month's level of backlog of orders (unfilled sales orders from customers) (units, not dollars) compared to the previous month is:	Higher/Same/Lower
Reason if higher or lower:	Free text
Indicate if the current month's level of new export orders (units, not dollars) (for delivery outside the U.S.) compared to the previous month is:	Higher/Same/Lower
Reason if higher or lower:	Free text
Indicate if the current month's approximate weighted average prices for the materials, commodities and services that you are ordering compared to the previous month is:	Higher/Same/Lower
Indicate if the current month's level of inventory of production inputs (including raw materials, intermediate products, MRO items, etc.) (units, not dollars) compared to the previous month is:	Higher/Same/Lower
Reason if higher or lower:	Free text
Do you perceive your customers' inventories of products they order from you, THIS MONTH, as:	Too High/About Right/Too Low/Do Not Know
Indicate if the current month's level of imports of production inputs (including raw materials, intermediate products, MRO items, resources/services, etc.) (units, not dollars) compared to the previous month is:	Higher/Same/Lower
Reason if higher or lower:	Free text
Indicate if the current month's level of employment (all personnel, not just supply management personnel) compared to the previous month is:	Higher/Same/Lower
Reason if higher or lower:	Free text
Indicate if the current month's supplier delivery performance for purchased commodities, materials and services compared to the previous month is:	Faster/Same/Slower
Reason if faster or slower:	Free text

Table 2: Survey Summary Statistics

Field	(1) Fraction W/ Text	(2) Mean Word Count	(3) Mean Word Count Cond. on Text
General Remarks	0.49	8.21	16.73
Production	0.27	1.47	5.53
New Orders	0.26	1.50	5.70
Backlog	0.19	1.20	6.46
Employment	0.01	0.07	5.10
Supplier Speed	0.12	0.92	7.72
Input Inventories	0.23	1.58	6.81
Exports	0.11	0.63	6.01
Imports	0.12	0.81	6.64
All Text (Appended)	0.68	16.40	24.27

Notes: This table provides summary statistics derived from the ISM survey. Column (1) reports the fraction of firm-month observations containing any text. Column (2) shows the mean word count across all firm-month observations, while column (3) shows the mean word count of only those responses containing any text. Each row corresponds to one of the various question types on the ISM survey.

Table 3: Summary Statistics

N=156	Mean	Std. Dev.	Min	Median	Max
Text Measures					
LM	-0.0091	0.0063	-0.0358	-0.0087	0.0055
Harvard	0.0000	0.0046	-0.0191	0.0003	0.0097
AFINN	0.0104	0.0109	-0.0327	0.0105	0.0343
Stability	-0.0016	0.0038	-0.0221	-0.0012	0.0092
FinBERT (v1)	-0.0649	0.1118	-0.4402	-0.0513	0.1472
FinBERT (v2)	-0.1193	0.1138	-0.5051	-0.0935	0.0891
TF-Small	0.0218	0.1188	-0.3923	0.0254	0.2565
Fine-Tuned BERT	-0.0936	0.1297	-0.5530	-0.0749	0.1986
Macro Variables					
IP Growth$_t$	-0.0438	0.7430	-3.3665	-0.0178	1.5712
ISM_PMI$_t$	52.9481	4.7944	34.5000	53.3500	60.9000
ISM_NewOrders$_t$	55.1205	6.7867	25.9000	55.9500	66.8000
ISM_Inventories$_t$	48.5212	4.3760	33.5000	49.2500	56.8000

Notes: Summary statistics for the variables used in the in-sample and out-of-sample analysis. LM, Harvard, AFINN, and Stability measure the average net sentiment when applying dictionary word counts of the Loughran and McDonald, Harvard, AFINN and Stability (Correa et al. 2019) word lists, respectively. FinBERT (v1) and FinBERT (v2) measure the average net sentiment of applying the FinBERT model from Araci 2019 and Yang et al. 2019, respectively. TF-Small and Fine-Tuned BERT are sentiment scores derived from a fine-tuned transformer and a fine-tuned BERT model using a sample of human-labeled ISM responses.

Table 4: In-sample Regression Results

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Dependent Variable: IP Growth$_t$								
		Dictionary Based Methods				*Deep Learning Methods*			
Text Measure	*Baseline*	LM	Harvard	AFINN	Stability	FinBERT (v1)	FinBERT (v2)	TF-Small	Fine-Tuned BERT
ISM_Sentiment$_t$		0.00737	0.0331	0.0604	0.103*	0.202*	0.204**	0.228**	0.152*
		(0.0604)	(0.0648)	(0.0656)	(0.0535)	(0.102)	(0.0829)	(0.113)	(0.0880)
ISM_PMI$_t$	0.0817***	0.0812***	0.0791***	0.0796***	0.0828***	0.0595***	0.0598***	0.0462**	0.0639***
	(0.0185)	(0.0180)	(0.0188)	(0.0179)	(0.0184)	(0.0184)	(0.0168)	(0.0221)	(0.0181)
IP Growth$_{t-1}$	-0.0281	-0.0293	-0.0327	-0.0427	-0.0467	-0.0595	-0.0604	-0.0472	-0.0531
	(0.0957)	(0.0965)	(0.0971)	(0.0969)	(0.0916)	(0.0947)	(0.0950)	(0.0932)	(0.0974)
IP Growth$_{t-2}$	0.0360	0.0349	0.0305	0.0206	0.00231	-0.00653	-0.0174	0.0190	0.00948
	(0.104)	(0.103)	(0.103)	(0.103)	(0.0989)	(0.107)	(0.107)	(0.104)	(0.104)
IP Growth$_{t-3}$	0.00789	0.00651	0.00255	-0.00732	-0.0166	-0.0354	-0.0372	-0.0125	-0.0221
	(0.104)	(0.106)	(0.104)	(0.106)	(0.104)	(0.109)	(0.108)	(0.105)	(0.107)
Observations	156	156	156	156	156	156	156	156	156
R-squared	0.288	0.288	0.289	0.292	0.304	0.315	0.314	0.314	0.303

Notes: This table reports in-sample regressions of the month-to-month percentage change of industrial production on a set of real-time predictors of IP from 2007m1-2019m12. $ISM\_Sentiment$ is a text measure of the survey response sentiment using either dictionary-based methods (columns 2-5), transfer learning of financial BERT models (columns 6-7), or fine-tuned models trained on a random selection of human-labeled ISM survey responses (columns 8-9). ISM_PMI is the monthly diffusion index of PMI released by the ISM at the beginning of the month. IP_Growth with lag t-1, t-2, or t-3 is the once, twice, or three-times revised estimate released in month t-1, t-2, or t-3, respectively.

Table 5: Out-of-sample Regression Results

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)		
		In-Sample: 2007M1-2017M12									
		Out-of-Sample: 2018M1-2019M12									
		Dictionary Based Methods				*Deep Learning Methods*				Controls Included	
Text Measure	*Baseline*	LM	Harvard	AFINN	Stability	FinBERT (v1)	FinBERT (v2)	TF-Small	Fine-Tuned BERT	IP Lags	ISM Variables
OOS MSE	0.437	1.14%	-0.23%	-1.60%	-10.76%	-7.32%	-12.81%	-8.47%	-9.15%	3 Lags	PMI
DM Test P-Value		0.161	0.525	0.678	0.089	0.184	0.088	0.101	0.016		
OOS MSE	0.42	1.43%	-0.48%	-2.38%	-11.19%	-9.29%	-13.57%	-13.57%	-9.76%	-	PMI
DM Test P-Value		0.121	0.389	0.47	0.04	0.066	0.036	0.036	0.004		
OOS MSE	0.339	-1.18%	5.60%	3.54%	4.72%	-1.18%	0.59%	6.49%	-9.14%	3 Lags	-
DM Test P-Value		0.213	0.385	0.227	0.475	0.828	0.906	0.426	0.029		
OOS MSE	0.413	1.69%	0.00%	0.00%	-6.05%	-2.42%	-5.81%	-0.97%	-3.39%	3 Lags	New Orders
DM Test P-Value		0.019	0.952	0.968	0.044	0.308	0.04	0.708	0.000		
OOS MSE	0.435	2.07%	-0.46%	-0.69%	-7.13%	-3.91%	-7.59%	-4.14%	-3.91%	3 Lags	PMI, New Orders, Inventories
DM Test P-Value		0.023	0.504	0.824	0.048	0.129	0.023	0.1148	0.000		

Notes: This table reports out-of-sample mean squared errors of regressions of month-to-month percentage change of industrial production on a set of real-time predictors of IP from 20018m1-2019m12. The text measures represent the survey response sentiment using either dictionary-based methods (columns 2-5), transfer learning of financial BERT models (columns 6-7), or fine-tuned models trained on a random selection of human-labeled ISM survey responses (columns 8-9). PMI, New Orders, and Inventories are monthly diffusion indices released by the ISM at the beginning of the month. The 3 lags are the IP_Growth lags at time t-1, t-2, or t-3 revised once, twice, or three-times, respectively. The P-values are calculated using the Diebold-Mariano out-of-sample error statistics.

Table 6: Average Net Positive Scores

Positive Words	Score	Negative Words	Score
negotiated	0.068	unstable	-0.455
improve	0.068	insufficient	-0.358
good	0.070	fragile	-0.344
gaining	0.072	inconsistent	-0.325
steady	0.073	questionable	-0.283
stronger	0.074	restricted	-0.262
strong	0.075	weaken	-0.250
robust	0.075	poor	-0.238
enjoying	0.075	shortage	-0.234
lab	0.078	hurting	-0.226
protecting	0.083	difficulties	-0.203
cleared	0.085	weak	-0.200
efficient	0.085	depression	-0.197
improvement	0.089	shortages	-0.197
improved	0.097	instability	-0.195
improving	0.099	delays	-0.195
booming	0.109	depressed	-0.194
helps	0.120	slipping	-0.194
excellent	0.129	declining	-0.190
brisk	0.135	reluctance	-0.188

Notes: Words are those with the most positive and most negative scores, among words appearing more than 5 times in the data. The "score" is the net positive probability from the Shapley decomposition: The average marginal contribution of the word toward a positive classification, minus the average marginal contribution towards a negative classification.

Table 7: Words with Lowest Sentiment in 2007:Q1-2009Q1 Comparison

affected	decreasing	horrific	poor	stalled
affecting	delay	hurricane	population	strike
bad	delayed	hurt	problem	toxic
bankrupt	delaying	hurting	problems	trough
bleak	delays	inability	quo	ugly
bleeding	depressed	inflation	ra	uncertainty
cancellation	depression	instability	ravaged	volatile
cancelled	deteriorating	lack	recession	weak
cannot	difficult	less	reduced	weaken
causes	difficulty	loss	reducing	weakening
causing	disruption	lost	reduction	weaker
ceased	down	low	reductions	weakness
closed	downward	lower	retirement	worried
closure	drain	lowered	retreating	worse
collapse	drop	lowering	severe	worst
concerns	dropped	mortgage	severely	##ag
conflicting	dropping	negative	shortage	##air
conservative	excessive	negatively	shortages	##gis
crisis	falling	none	shrinking	##hwa
cuts	fell	nothing	slow	##sett
decline	fewer	offs	slowed	##sma
declines	flood	overs	slower	##une
declining	forcing	paralyzed	slowing	
decrease	grave	plague	slug	
decreased	hampered	plum	slump	

Notes: Tokens in the lowest 5 percent of average sentiment, alphabetical order. Tokens prepended with "##" are word fragments, which appear in the comments as part of a larger word.

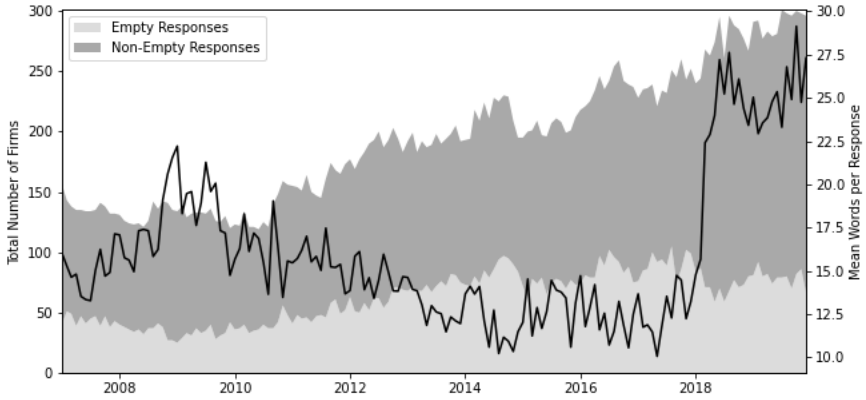
Table 8: Words with Highest Sentiment in 2007:Q1-2009Q1 Comparison

active	emphasis	healthy	ok	stable
activities	everybody	help	opened	steady
added	evidence	hopefully	opportunities	strategy
adjusting	excellent	implemented	optimistic	strengthening
advantage	expand	improve	outlook	strong
airplanes	expanded	improved	picking	stronger
arrived	expanding	improvement	pickup	supporting
avail	expansion	improvements	positive	techniques
balanced	extended	improves	preparing	traction
best	extending	improving	pretty	unchanged
better	favorable	included	priority	upbeat
booming	filled	increase	promising	warm
boost	focus	increased	pursuing	watching
bracing	gaining	increasing	ready	worked
bright	glad	initiated	realization	##ability
brighter	goal	initiative	rebuild	##cing
builds	good	initiatives	relaxation	##hg
busy	gradually	introduced	responding	##my
completed	great	launches	risen	##rry
completing	greater	launching	robust	##screen
connected	grew	lots	selected	##yas
consistent	grow	lust	soared	
controlling	growing	matched	solid	
decent	grown	matching	solutions	
effort	growth	medical	stability	

Notes: Tokens in the highest 5 percent of average sentiment, alphabetical order. Tokens prepended with "##" are word fragments, which appear in the comments as part of a larger word.

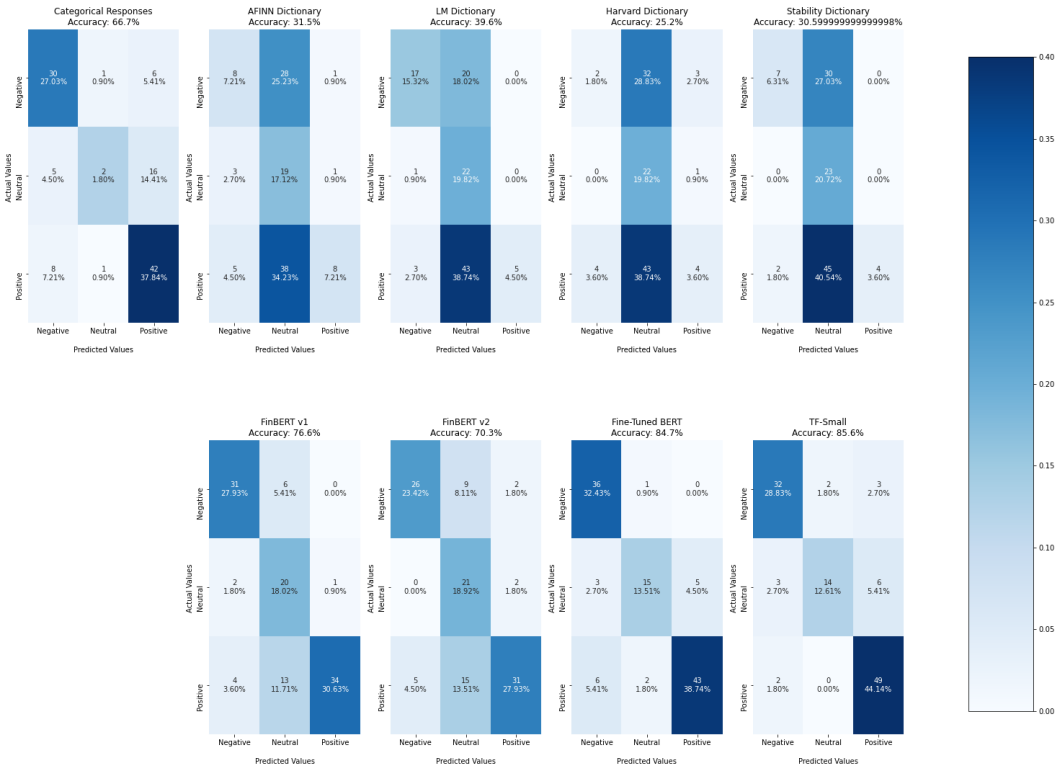
Figures

Figure 1: ISM Survey Responses



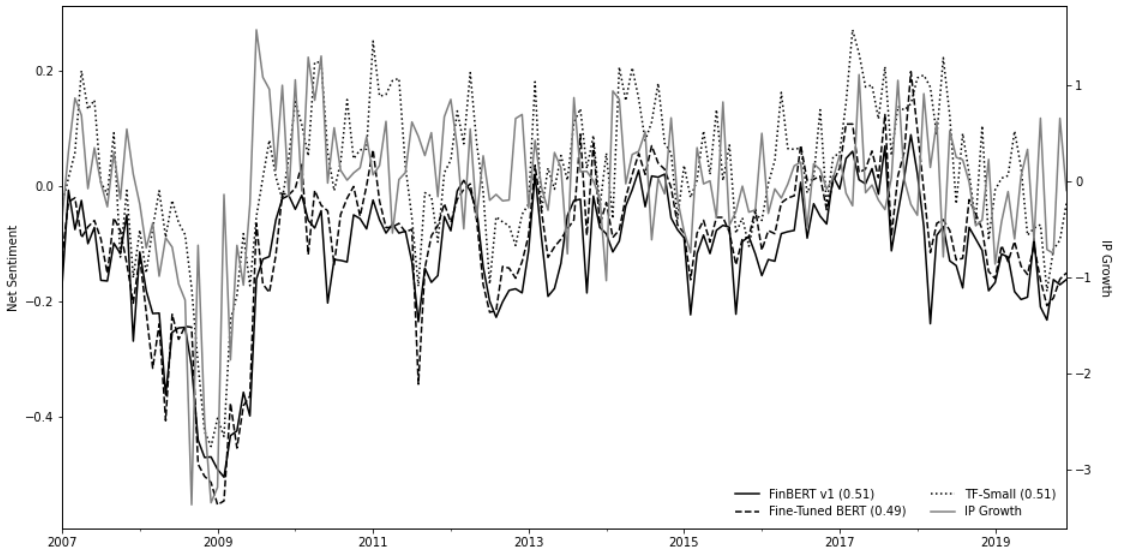
Notes: This figure shows the total number of firms and the word counts for the ISM survey responses. [Left Axis] The light (dark) grey region shows the total number of firms that provided empty (non-empty) responses on their monthly response. The total number of firms is the height of the light and dark grey regions. [Right Axis] The black line shows the mean number of words per response across all respondents for a given month.

Figure 2: Confusion Matrices & Accuracy Scores



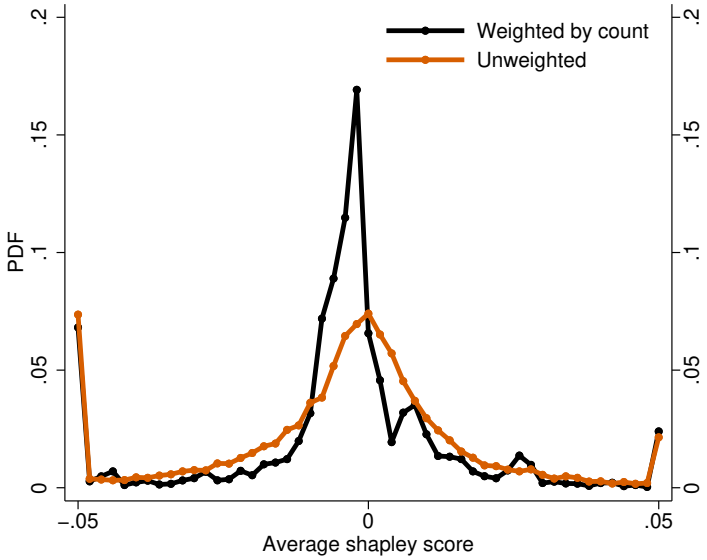
Notes: This figure shows the confusion matrix for eight manufacturing sentiment measures applied to the training dataset of manually labelled ISM survey responses. The rows of each matrix refer to the actual values, while the columns refer to the predicted values. Values along the diagonal are correctly classified, while values on the off-diagonals are incorrect. The shaded color refers to the percentage of responses within a given cell, according to the heatmap legend on the right.

Figure 3: Industrial Production and Sentiment



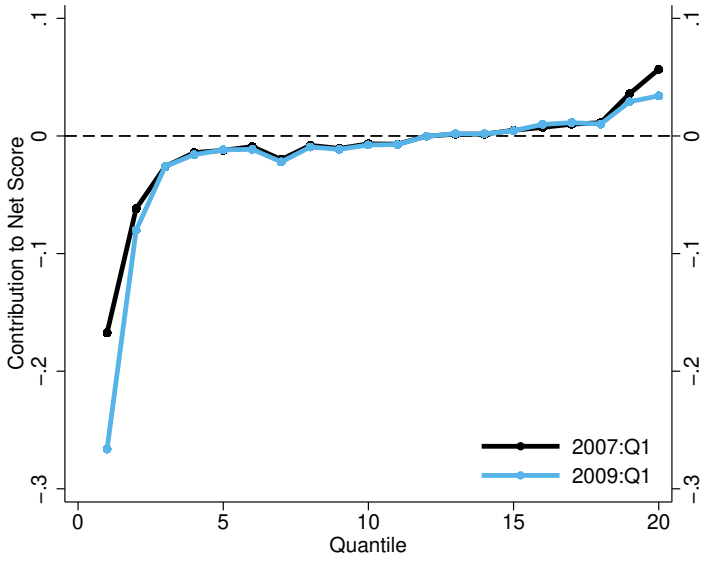
Notes: This figure shows various manufacturing sentiment measures alongside IP Growth (grey). FinBERTv1 is a widely available BERT models trained on financial text. TF-Small and Fine-Tuned BERT consist of a baseline transformer model and a widely available BERT model, respectively, both of which have been fine-tuned on a training dataset of human-labeled ISM survey responses. Correlations to IP Growth are provided in parentheses.

Figure 4: Token PDFs



Note: Distribution of tokens across Shapley scores. For this graph, Shapley scores are Winsorized at -0.05 and +0.05. "Unweighted" gives the distribution of unique tokens by score, "Weighted" gives the distribution weighted by number of appearances in the data.

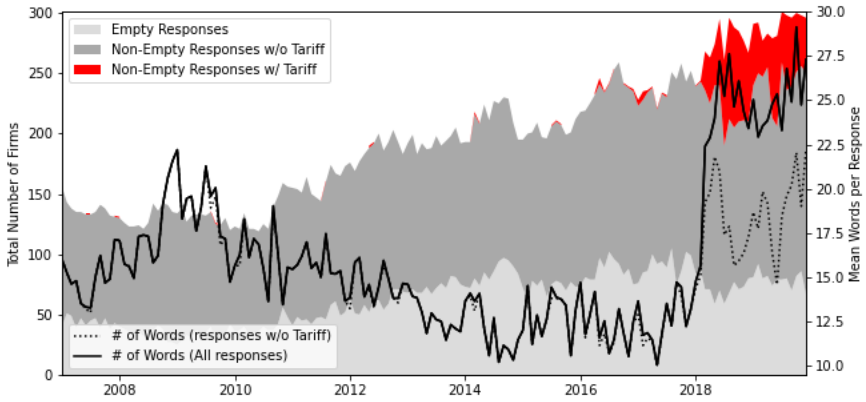
Figure 5: Sentiment Contributions by Quantile



Note: A token's contribution to the net aggregate score is their own score multiplied by the number of occurrences, and scaled to account for the number of comments in the relevant period. Tokens that appear in either 2007:Q1 or 2008:Q1 are divided into 20 quantiles, based on their average Shapley score. Contributions are summed within each quantile.

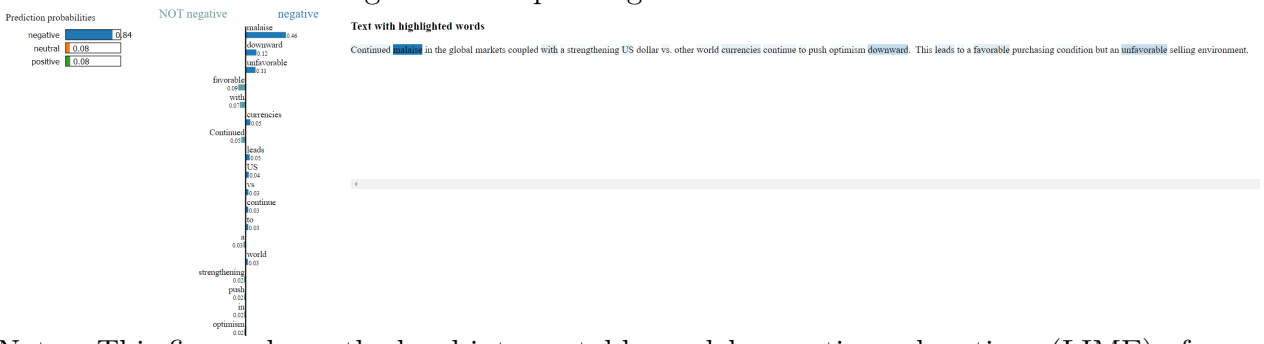
Appendix

Figure A1: What Explains the 2018 Increase in Text Responses?



Notes: This figure shows the total number of firms and the word counts for the ISM survey responses. [Left Axis] The light (dark and red) grey region shows the total number of firms that provided empty (non-empty) responses on their monthly response. The total number of firms is the height of the light and dark grey regions. The red region highlights the number of firms that included the word "tariff" in their response. [Right Axis] The solid (dotted) black line shows the mean number of words per response across all respondents (excluding responses using the word "tariff") for a given month.

Figure A2: Explaining Fine-Tuned BERT



Notes: This figure shows the local interpretable model-agnostic explanations (LIME) of our Fine-Tuned BERT model. The leftmost panel shows the probability that the text example belongs to one of the three sentiment classifications. The middle panel illustrates the increase in probability that a given word has on the Negative classification. This technique can be applied to any individual observation to help explain which words trigger a particular sentiment classification.

7 Summary Text Methods

7.1 Dictionary Based Methods

A bag of words dictionary method is a mapping of the form $f : \mathbb{R}^V \to \mathbb{R}$ where $x^d \in \mathbb{R}^V$ is a V-dimensional vector and V represents the size of the set of the unique tokens across a corpus, S. The elements of x^d, e.g. $x^d_{w_i}$, represent the number of occurrences of the word w_i in document d.

For a bag of words method, we select a subset of the unique words across the corpus, $D \subset S$. Then, the function f is simply the sum of the elements in x^d, i.e. $f(x^d) = \sum_{w_i \in D} t_{w_i} x_{w_i}$. t_{w_i} represents the weight given to word w_i. Typically, for sentiment analysis, there are three weights: +1 for positive words, 0 for neutral words, and -1 for negative words inside of D.

7.2 BERT Models

This section describes the basics of BERT, one of the most popular transformer-based models. It is difficult to explain transformer-based models briefly, in part because they are fundamentally complex. Existing descriptions of these models are either very terse, assume extensive knowledge of deep learning terminology and history, or are vague. Our goal to provide a reasonably succinct overview of the architecture, accessible to someone not specialized in deep learning.

These models are called "transformers" because the input is transformed into a representation in a latent space.[11] This aspect of the architecture is not particularly unique; the main distinguishing feature of transformers is *attention*, a mechanism that allows the interpretation of words in a sentence to be influenced by the other words in the sentence.

Transformers gained popularity in part because they showed excellent performance on a wide variety of language tasks and, relatedly, the design allows for extreme parallelism.

Section 7.2.1 describes in detail the mechanics of what happens when text is fed to a BERT model used for sentiment classification (or more broadly, any type of classification). Section 7.2.2 goes over how BERT models are trained. Section 7.2.3 discusses how the BERT model is further trained and specialized ("fine-tuned") to perform specific tasks or use additional data.

[11] In the original transformer paper the application was machine translation. The input, in one language, was transformed ("endcoded") into an abstract representation and this was then "decoded" back into the second natural language. The BERT architecture only includes the encoding step, and classification or other tasks use the abstract representation as an input. GPT-like models are considered decoder-only models, which seek to generate the next word in a sequence using a representation of the sequence so far.

7.2.1 BERT at inference

BERT at inference can generally be defined in five steps. First, the input text is partitioned into its atomic unit, e.g. a word, in a process known as tokenization. Each token is represented in an abstract vector space that captures the syntactic and semantic meaning of the token. Second, the word order is taken into account using positional embeddings. Third, the model adjusts the attention it should place to other words in the sequence through the defining characteristic of transformers, known as the *attention mechanism*. Fourth, a normalization step concatenates the attention with the input embeddings. Lastly, the new representation of the input sequence is used for sentiment classification. We guide the reader through these five steps below.

Step 1: Creating the Input Embeddings

A transformer-based sentiment model can be defined as a mapping of a fixed number of tokens, L, such that $f_T : \mathbb{R}^{V \text{x} L} \rightarrow \mathbb{R}$, where the input x is a $V \times L$ matrix.

A *token* is a word, a part of a word or a single character. V is the size of the *vocabulary*, the tokens that are valid inputs.[12] The columns of x are dummy vectors of size V, with element i equal to 1 if the word in the $i-th$ position is equal to w_i, and zero otherwise. Many pre-trained BERT models fix L, the sequence length, at 512 tokens. If a sequence contains less than 512 tokens, then the remaining sequences are "padded", in other words replaced with a special "end of sequence" token that will mask any parameters associated with those positions. If a sequence has more than 512 tokens, only the first 512 would be used.

Transformers, like most NLP methods, represent words as vectors, called embeddings. In large-scale, general versions of BERT, such as the base version released by Meta,[13] the word (token) is represented as a 768-dimensional vector. The high dimensionality should help capture the fact that words' meanings have many dimensions, so two words can be be similar in many ways but still distinct along important dimensions.

At inference time the embeddings are fixed. The first step of f_T is to convert the $V \text{x} L$ input into a $N \text{x} L$ matrix, where each token indicator column (of length V, the size of the

[12] BERT has a vocabulary of 30,522 tokens. These tokens include most common words, and "token" is sometimes used interchangeably with "word". But, importantly, the vocabulary also includes many word parts, such as common word endings, and all single characters. Thus BERT can process any text, since unfamiliar words can be built up from word fragments and single characters.

[13] https://github.com/google-research/bert/blob/master/README.md

vocabulary) is converted into a length N word embedding vector. Define the NxL matrix as x'.

Step 2: Adjust to Generate Positional Embeddings

Transformer models do not inherently account for the order of the inputs anywhere in their architecture, a characteristic that is critical for understanding the meaning of text. Adding an index number of the input token (e.g. 1 for the first token, 2 for the second token, etc.) would create two difficulties. First, this method leads to unbounded positional adjustments. Second, the model may not be able to generalize for sequence lengths that are rarely seen, especially longer sequences. The model could see plenty of first word adjustments, second word adjustments, etc. but larger values would become rarer. The typical solution to account for positions is to use sine and cosine functions. For input token x'_k, an N-dimension vector - p^k- is generated. For $0 \leq i < N/2$, :

$$\begin{aligned} p^k_{2i} &= sin(\frac{k}{10000^{\frac{2i}{N}}}) \\ p^k_{2i+1} &= cos(\frac{k}{10000^{\frac{2i}{N}}}) \end{aligned} \tag{3}$$

where p^k_i is the i-th index of p^k and N is the dimension size of the target embedding.

We adjust the column vectors of x' for their position by adding p to x'. Call the adjusted matrix $y \in \mathbb{R}^{N\text{x}L}$

Step 3: Attention Mechanism

Next we enter the transformer block. This is a mapping $f : \mathbb{R}^{N\text{x}L} \to \mathbb{R}^{N\text{x}L}$. Note that the output and input are the exact same size. This step of the transformer model is arguably the most important as the final representation of the word vectors captures well the meaning of the text.

We begin by creating a set of key, value, and query matrices. This step mimics a look-up table in a database table. They are defined as follows:

$$\begin{aligned} K(y_i) &= W_k y_i \\ V(y_i) &= W_v y_i \\ Q(y_i) &= W_q y_i \end{aligned} \tag{4}$$

where $y_i \in \mathbb{R}^N$ (a column vector from the input y) and $W_k, W_v, W_q \in \mathbb{R}^{M\text{x}N}$. Ultimately,

the resulting vectors $K(y_i), V(y_i), Q(y_i) \in \mathbb{R}^M$ are transformations of the input vector, y_i. This can be thought of as projecting the individual vector y_i into into an abstract M-dimensional space. An LxL matrix is then created:

$$\alpha = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1L} \\ \vdots & \ddots & \vdots \\ \alpha_{L1} & \dots & \alpha_{LL} \end{bmatrix} \tag{5}$$

where $\alpha_{i,j} = \text{softmax}_j(\frac{Q(y_i) \cdot K(y_j)}{\sqrt{M}})$ (i.e. the rows of the α matrix sum to 1). Essentially, $\alpha_{i,j}$ measures how similar the query is (a transformation of the i-th word in consideration) to the other keys (a transformation of the words in the sentence).

The vector for the i-th vector is then weighted depending on the attention of that word with the other words in the sentence.

$$u'_i = W_0 \sum_{j=1}^{L} \alpha_{i,j} V(y_j) \tag{6}$$

where $W_0 \in \mathbb{R}^{N\text{x}M}$ and $u'_i \in \mathbb{R}^N$. This assumes there is just one head. However, we can have multiple heads such that equations (4), (5), and (6) are repeated with H different sets of parameters. For example, for head h, the W matrices in (4) will be different: $W_{k,h}.W_{v,h},$ and $W_{q,h}$. This will lead to $\alpha_{\mathbf{h}}$ in (5), and (6) will become:

$$u'_i = \sum_{h=1}^{H} W_{0,h} \sum_{j=1}^{L} \alpha_{i,j}^{(h)} V^{(h)}(y_i) \tag{7}$$

with $W_{0,h} \in \mathbb{R}^{N\text{x}M}$.

The last step of the attention mechanism is to add back the resulting matrix u'_i from (7) back to the input vector y_i, and then pass the resulting vector through a layer normalization function, which is analogous to a standard normalization procedure but slightly adjusted with a different scaling and shifting parameter.[14]

[14]The layer normalization function has two hyperparameters, γ and β, and is defined as follows: $\text{LayerNorm}(x; \gamma, \beta) = \gamma * \frac{x-\mu}{\sigma} + \beta$

$$u_i = \text{LayerNorm}(y_i + u_i') \tag{8}$$

Step 4: Feed Forward and Normalize

Next the resulting vector, u_i, is passed to a ReLU network, then added to itself, and finally normalized once more:

$$z_i' = W_2 \text{ReLU}(W_1 u_i) \tag{9}$$

$$z_i = \text{LayerNorm}(u_i + z_i') \tag{10}$$

where $W_1 \in \mathbb{R}^{P\text{x}N}$ and $W_2 \in \mathbb{R}^{N\text{x}P}$. The final vector $z_i \in \mathbb{R}^N$ is the transformed input vector y_i that accounts for the position of the i-th word and the attention the word emits and receives from other words in the sequence.

Step 5: Sentiment Classification

The last step entails a mapping $f : \mathbb{R}^{N\text{x}L} \to \mathbb{R}$. Typically, this is a neural network that takes as input a matrix and outputs a probability distribution across 3 categories: positive, negative, and neutral.

7.2.2 Training

As with most deep learning models, BERT is estimated using stochastic gradient descent. Model weights are adjusted using a learning rate, λ, such that $w_{i+1} = w_i - \lambda \frac{\delta L}{\delta w_i}$, where L is the loss function. If λ is too large, updates may exceed w_i and the optima may be missed. Setting λ too small may lead to smaller adjustments and more time needed for convergence. To accelerate the process and improve the efficiency of finding optimum weights, an extension of gradient descent, known as Adaptive Moment Estimation (or the ADAM optimizer), is typically used.

For financial text sentiment classification, two popular BERT models have been pre-trained on large corpi of data and are publically available: Huang et al. (2022) (which we refer to as FinBERTv1) and Araci (2019) (which we refer to as FinBERTv2). FinBERTv1 was trained on nearly 10,000 sentences from SEC filings, equity reports, and earnings conference call transcripts that were hand labelled for sentiment. FinBERTv2 was trained

on nearly 5,000 randomly selected sentences from financial news articles, and nearly 1,000 financial news tweets, all of which were manually labelled for sentiment.

7.2.3 Fine-Tuning

We fine-tune BERT by first creating a dataset of responses that were hand-labeled for sentiment. We format the ISM survey responses the firm-month-question level and randomly select 2,000 text responses. Each response was classified for sentiment by two Federal Reserve economists using the following question as a guide: "Is this comment consistent with manufacturing IP rising month over month?" The classifications were either positive, neutral, or negative. We keep only 1,543 responses for which both economists agreed on the sentiment.

We split our sample such that 90% is used for fine-tuning, and 10% is leftover for an unseen test set for sentiment model comparisons (i.e. is never used for the training). We use the training data to train two types of models. The first uses a publicly available pre-trained model trained on a large corpus of English. We fine-tune the last layer (the sentiment layer) of this model to create *Fine-Tuned BERT*. Second, we train a plain vanilla transformer model from scratch using a simple architecture (with embedding dimensions of size 12-16). We call this model simply the *TF-Small* model (TF for 'transformer'). Note that for the *TF-Small* model, we are estimating the entire attention mechanism weights, whereas for the *Fine-Tuned BERT*, we are further tuning the attention weights that were pre-trained on a large dataset.