[File Link](https://www.dropbox.com/s/tr4le96l00xyp7y/BSTW%20Earnings%20Announcement%20Dataset.csv?dl=0) – This is a link to the DropBox Folder with the file in .CSV and .DTA forma.

This document provides a brief discussion of the variables appearing in the Earnings Announcement dataset. Please cite Bentley, Stubbs, Tian, and Whited (2023) when using this dataset. The document also includes Appendix A from that manuscript describing the Python code used to search the earnings announcements and accuracy tests performed on the data. Please reach out to the authors with any questions about the dataset.

**Dataset Description:**

gvkey – gvkey from Compustat

datadate – Datadate from Compustat

link – URL of Earnings Announcement

sequence – Order of the document in the URL that classified as the text Earnings Announcement

docname – Name of document classified as text Earnings Announcement

docdescript – Description of document classified as the text Earnings Announcement

totaldoccount – number of documents in the filing

totalwords – number of words in the text of the Earnings Announcement excluding tables

**Prefix: For traits, the prefix identifies to what the trait relates**

document – traits of the entire document

earnings – traits of earnings

revenue – traits of revenue

nongaapearnings – traits of non-GAAP earnings

ebit – traits of earnings before interest and taxes

ebitda – traits of earnings before interest, taxes, depreciation, and amortization

opcashflow – traits of operating cash flows

gross profit – traits of gross profit

gross margin – traits of gross margin

dividend – traits of dividends

**Traits**

table – number of times the metric appears in a table in the document

center – number of times the metric appears centered in the document

bullet – number of times the metric appears in a bullet point in the document

bold – number of times the metric appears bolded in the document

sentences – number of unique sentences discussing the metric in the document

fog – average fog index of sentences discussing the metric in the document

passive – number of sentences discussing a metric which include passive voice

negate – number of sentences discussing a metric which include negation

count – number of mentions of a metric

words\_before – number of words preceding the first mention of the metric

appears – indicator of the metric appears in the Earnings Announcement

Appendix A from Bentley, Stubbs, Tian, and Whited (2023) describing Python Code and Accuracy Tests

**Appendix A*.* Python Code and Accuracy Tests**

Our full code is too extensive to provide here, but more information can be obtained by contacting the first author. However, we summarize the search technique in this appendix to provide some explanations of our approach. We then compare our code to data from hand-collection to validate the accuracy of the code.

*Identifying the Earnings Announcement*

Approximately 81% of filings contain just two documents. In most instances, these two documents include the official 8-K and a single EX-99 attachment containing the press release (i.e., EA), making it easy to identify the correct document. In other instances, the official 8-K includes the EA while the EX-99 attachment contains supplemental financial statements. The remaining 19% of filings contain three or more documents. When a filing contains multiple documents, we employ the following decision criteria. If a single step removes all documents, then we skip that step. If a step eliminates all but one document, then we end the decision tree at that step. First, we focus on documents that are identified as EX-99 or, if there are no EX-99 documents, documents that are identified as a press release. Next, we limit our search to text-based documents (i.e., we remove files that have an extension of .ppt, .pptx, .xls, .xlsx, .pdf, or .jpg). Third, we remove supplemental materials (including tables) and letters to shareholders. Fourth, we remove conference calls. Fifth, we keep only documents that self-identify as a press release. Sixth, we remove documents that appear to be non-EA press releases, such as announcements of mergers, acquisitions, purchase agreements, share repurchases, and director/executive appointments. Seventh, we restrict the search to documents that have “announce” or “report” in the first 1,000 characters. Finally, we keep the document with the highest total number of earnings words. At the end of the process, if the resulting document has zero earnings words outside of a table, then we restart the search, looking for the document with highest number of earnings words not in a table.

*Search Technique*  
Our code takes five broad steps.[[1]](#footnote-1) First, we clean up the document by dealing with tables, HTML, punctuation, capitalization, parentheses, alternative spellings, and plurals. In some instances, we write custom code to deal with issues specific to our setting. For example, our code separates hyphens used as bullet points from hyphenated words and hyphens used as a negative symbol. We also remove hyphens in phrases that are sometimes hyphenated (e.g., “earnings-per-share” is converted to “earnings per share”). Similarly, we adjust for phrases that sometimes include a space (e.g., “non gaap” vs “non-gaap” vs “nongaap”). We use Python packages (e.g., beautifulsoup, textstat, liwc) to deal with more generic situations (e.g., stripping out html).

Second, we make some basic, recurring substitutions to make subsequent code more efficient. Parentheses indicate variants searched for, a blank cell in the “Replacement Phrase” column means that any matches of the search term are deleted from the text.

|  |  |
| --- | --- |
| **Search term** | **Replacement Phrase** |
| profitable | \_nonearnings\_ |
| fully diluted |  |
| diluted |  |
| common |  |
| basic |  |
| net (income/loss/earnings) | earnings |
| earned | earnings |
| sales | revenue |
| income | earnings |
| earnings tax | tax |
| comprehensive earnings | earnings |

Third, we search for phrases that should be excluded (i.e., rule out false positives). An abbreviated list is included below, although we note that steps 1 (cleaning up the html) and 2 (basic substitutions) greatly increase the variation in terms. Parentheses indicate variants or optional words. Underscores surrounding a word indicate a phrase that was previously identified (e.g., “\_grossprofit\_” indicates searching for any “grossprofit” construct previously identified). Note that some elements of this code run after step 4, to allow variations from step 4 code.

|  |  |
| --- | --- |
| **Abbreviate Set of Search Terms** | **Construct** |
| profitable | nonearnings |
| loan loss | nonearnings |
| cost of (sales/revenue) | nonearnings |
| (earnings/sale/revenue) (release/conference call/press release/announcement/statement/schedule) | earningsrelease |
| (schedule/statement) of (earnings/revenue/revenue and earnings/earnings and revenue) | earningsrelease |
| earnings from discontinued operations | nonearnings |
| (other/segment/investment) earnings | nonearnings |
| retained \_earnings\_ | nonearnings |
| (same/comparable) store \_revenue\_ | nonearnings |

Fourth, we search for constructs of interest (including some constructs not analyzed in the paper).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Abbreviate Set of Search Terms** | **Construct** |  | **Abbreviate Set of Search Terms** | **Construct** |
| gross (profit/loss) | grossprofit |  | \_earnings\_ before interest and tax | ebit |
| gross (earning/income/) (margin/percent) | grossmargin |  | ebit | ebit |
| \_grossprofit\_ (margin/percent) | grossmargin |  | operating \_earnings\_ | ebit |
| (revenue/sales) | revenue |  | \_earnings\_ (from operations/before tax from continuing operations) | ebit |
| (earnings per share/eps) | eps |  | \_earnings\_ excluding | nongaapearnings |
| (profit/loss/lost/earnings) | earnings |  | (normalized/proforma/pro forma/core/nongaap/adjusted) \_earnings\_ | nongaapearnings |
| (cash flow from/cash from/cash used in/funds generated by/funds provided by/funds from) (operating activities/operation) | opcashflow |  | (normalized/proforma/pro forma/core/nongaap/adjusted) \_revenue\_ | nongaaprevenue |
| operating cash flow | opcashflow |  | \_revenue\_ excluding | nongaaprevenue |
| free cash flow | freecashflow |  | (normalized/proforma/pro forma/core/nongaap/adjusted) \_eps\_ | nongaapeps |
| dividend | dividend |  | \_eps\_ excluding | nongaapeps |
| \_earnings\_ before interest tax depreciation and amortization | ebitda |  | (\_eps\_/\_earnings\_/\_revenue\_/  \_nongaapearnings\_/\_nongaapeps\_/ \_nongaaprevenue\_) guidance | guidance |
| ebitda | ebitda |  |  |  |

Finally, we identify the textual highlighting of each metric. Conceptually, a metric is highlighted if it is formatted in a way that draws attention to it. We establish two criteria for a metric to be highlighted. First, it needs to be early in the document. For purposes of our Python code, we define “early” as the first 300 words or first quarter of the document (whichever is longer). Second, the metric needs to be distinguished from plain text. Through anecdotal reading of filings, we noted that the most common ways to highlight text were through bolding, bullet lists, and centering. Conceptually, underlining and italics could also be used. However, we observed very little use of underlining, and italics were used almost exclusively for safe harbor disclosures. As such, we do not code those two metrics.

After limiting each filing to the first 300 words or first quarter of the document (whichever is longer), our code searches for blocks of text with different formatting styles and then checks if any of our metrics (identified in steps 1-4) are included in that block of text. We identify centered, bolded, or bulleted text using a combination of html tags including <B>, <STRONG>, <CENTER>, <FONT>, <DIV>, <SPAN> and <UL>. We also search for bullet lists formatted using tables or text with negative bottom margins, as well as text formatted using hard-coded characters (e.g., a hyphen for a bullet point, a series of spaces to center information) for documents that do not use html.

*Accuracy Rates*  
To test the accuracy of our code, we had two research assistants who did not work on the Python manually code EAs in three steps. First, they checked whether Python identified the correct EA document, as described in Section 3.2 of the text. We identified the correct document in 197 out of 200 instances (98.5% accuracy).

Second, for a subsample of 197 filings where we identified the correct document, the research assistants code whether each of our metrics of interest was disclosed in the document. Two authors who did not work on the python code then reconciled any disagreement among the research assistants without looking at the dataset.

The table below presents accuracy rates for each metric, based on whether the Python code accurately identified the metric in the press release. We walk through revenue to demonstrate how to interpret the table. Of the 197 EAs, our code identifies a revenue mention in 189 and no revenue mention in 8. The research assistants identified a revenue mention in 188 of the 189 “Yes” revenue observations (99.47%) and identified no revenue mention in 5 of the 8 EAs where the Python code did not identify a revenue mention (62.50%). Thus, the overall accuracy was 97.97% for revenue. The code is least accurate at identifying EBIT (overall accuracy = 86.8%), operating cash flows (overall accuracy = 87.8%) and non-GAAP earnings (overall accuracy = 88.8%). Overall accuracy for all other variables is greater than 93%, and the average accuracy across all variables is 94.10%.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **Identified in Dataset** | **N** | **N Correct** | **Percent Correct** |
| Earnings | No | 0 | N/A | N/A |
| Earnings | Yes | 197 | 197 | 100.00% |
| Overall Earnings |  | 197 | 197 | 100.00% |
| Revenue | No | 8 | 5 | 62.50% |
| Revenue | Yes | 189 | 188 | 99.47% |
| Overall Revenue |  | 197 | 193 | 97.97% |
| Non-GAAP | No | 113 | 99 | 87.61% |
| Non-GAAP | Yes | 84 | 76 | 90.48% |
| Overall Non-GAAP |  | 197 | 175 | 88.83% |
| EBIT | No | 44 | 37 | 84.09% |
| EBIT | Yes | 153 | 134 | 87.58% |
| Overall EBIT |  | 197 | 171 | 86.80% |
| EBITDA | No | 144 | 144 | 100.00% |
| EBITDA | Yes | 53 | 52 | 98.11% |
| Overall EBITDA |  | 197 | 196 | 99.49% |
| Operating Cash Flows | No | 98 | 89 | 90.82% |
| Operating Cash Flows | Yes | 99 | 84 | 84.85% |
| Overall Operating Cash Flows |  | 197 | 173 | 87.82% |
| Gross Margin | No | 149 | 139 | 93.29% |
| Gross Margin | Yes | 48 | 46 | 95.83% |
| Overall Gross Margin |  | 197 | 185 | 93.91% |
| Gross Profit | No | 126 | 125 | 99.21% |
| Gross Profit | Yes | 71 | 68 | 95.77% |
| Overall Gross Profit |  | 197 | 193 | 97.97% |

Finally, for the subsample of metric-EA observations where the Python code indicated that a metric was disclosed, we evaluate whether the code correctly identified whether the metric was textually differentiated at least one time in the EA (i.e., bolded, bulleted, and/or centered). In the 197 EAs evaluated, we identified 894 mentions of the 8 metrics (sum of the “Yes” observations).[[2]](#footnote-2) In other words, our sample includes 197 EA-earnings, 189 EA-revenue, 84 EA-NG earnings, 153 EA-EBIT, 53 EA-EBITDA, 99 EA-operating cash flows, 48 EA-gross margin, and 71 EA-gross profit. Our research assistants performed an independent assessment of whether a given metric was bolded, bulleted, or centered at least once in a given document and we compared that to our dataset. Of the 153 metric-EAs which our code identified a bolded mention of the given metric, the research assistants identified a bold mention of that metric in 138 metric-EAs (90.20% correct). Conversely of the 894 metric-EAs, the Python code did not identify a bolding of the metric in 741 metric-EA observations. The graduate assistants identified no bolding of the given metric in 709 of these instances (95.68 percent). For example, a given observation would be an EA where the code did not identify any bolding of an earnings term. Our graduate assistants would evaluate that EA for whether earnings were bolded at least once in that EA. If there was no bolding of earnings terms (outside of tables), then this would be classified as a “correct” observation. The table below presents our accuracy rates for each formatting style, based on whether our Python code said that a particular formatting style was used for a particular metric. The code has an overall accuracy of at least 90% for each formatting style.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Format Attribute** | **Python Result** | **N** | **N Correct** | **Percent Correct** |
| Bold | 0 | 741 | 709 | 95.68% |
| Bold | 1 | 153 | 138 | 90.20% |
| Overall Bold |  | 894 | 847 | 94.74% |
| Bullet | 0 | 658 | 610 | 92.71% |
| Bullet | 1 | 236 | 207 | 87.71% |
| Overall Bullet |  | 894 | 817 | 91.39% |
| Center | 0 | 774 | 749 | 96.77% |
| Center | 1 | 120 | 108 | 90.00% |
| Overall Center |  | 894 | 857 | 95.86% |
| Highlight | 0 | 563 | 510 | 90.59% |
| Highlight | 1 | 331 | 300 | 90.63% |
| Overall Highlight |  | 894 | 810 | 90.60% |

1. The steps move back and forth in order to accommodate various specific phrases. For example, “cost of sales” needs to be classified as a “cost of goods sold” phrase before “sales” can be classified as a “revenue” phrase, which needs to occur before “non-GAAP sales” can be classified as an exempted phrase. [↑](#footnote-ref-1)
2. The total number of metric-EA observations was 1,576 (8 metrics \* 197 EAs). However, we exclude metric-EAs where a given metric was not mentioned (i.e., “No” in the prior table) to make accuracy rates more conservative. This is conservative because a metric that is not mentioned is inherently not textually differentiated. If we include these observations, our overall accuracy rates increase to 94.75% for bulleting, 96.94% for bolding, 97.56% for centering, and 94.31% for overall highlighting. [↑](#footnote-ref-2)