SPread: Automated financial metric extraction and spreading tool from earnings reports

Armineh Nourbakhsh
armineh.nourbakhsh@spglobal.com
S&P Global
New York, NY

Mohammad M. Ghassemi
ghassem3@msu.edu
Michigan State University
East Lansig, MI

Steven Pomerville
steven.pomerville@spglobal.com
S&P Global
New York, NY

ABSTRACT

In this paper, we present **SPread**, an automated financial metric extraction and spreading tool from earnings reports. The tool is created in a document-agnostic fashion, and uses an interpolation of tagging methods to capture arbitrarily complicated expressions. SPread can handle single-line items as well as metrics broken down into sub-items. A validation layer further improves the performance of upstream modules and enables the tool to reach an F1 performance of more than 87% for metrics expressed in tabular format, and 76% for metrics in free-form text. The results are displayed to end-users in an interactive web interface, which allows them to locate, compare, validate, adjust, and export the values.

1 INTRODUCTION

Every quarter, publicly-traded companies in the U.S. post an earnings report to the Security and Exchange Commission (SEC) as a form 8-K [1]. The report includes a breakdown of major performance indicators of the company over the previous quarter. Often, year-to-date numbers are also included, as well as numbers from comparable quarters from previous years. The company might also disclose guidance for future quarters or years.

In contrast to quarterly and annual reports such as forms 10-Q and 10-K, the earnings reports often don't have any section-structure imposed on them [2]. They usually begin with a summary of the business's performance, sometimes including a series of bullet points detailing metrics such as Revenue/Sales, Earnings Per Share, sector-specific or company-specific metrics, followed by tables with detailed breakdowns of GAAP or non-GAAP metrics [1]. Even though these reports are shorter and less detailed than corresponding 10-Q and 10-K forms, they are released up to a few days in the advance, and are thus important to credit analysts and market researchers from a timeliness perspective.

In this paper, we present **SPread**, a system that ingests earnings reports in real-time, classifies them, and extracts financial metrics from them. The extracted metrics are validated according to historical trends as well as numeric and positional cues. Finally, the metrics are displayed on an interactive web interface that allows the users to review and adjust them as needed. To the best of our knowledge, this is currently the first such tool that performs all extraction and validation steps in a fully automated fashion and in real-time.

[1] https://www.sec.gov/fast-answers/answersform8khtm.html

[2] The SEC requires companies to file their financial statements using the Inline eXtensible Business Reporting Language (iXBRL) by June 2021 [2]. This will impose a structured tagging system on the HTML document that ties each metric to a fixed taxonomy. However, SPread is designed in such a way to be extensible to financial reports outside of the SEC universe, as is detailed in Section 3. SPread can also be applied to historical reports that do not have any structured metadata.

2 BACKGROUND AND CHALLENGES

In the academic literature, any problem of tagging spans of text is closely associated with the problem of Named Entity Recognition (NER) for which many sequence modeling methods have been applied with great success [3, 5]. However, there are major differences between the NER challenge and the challenge of extracting financial metrics from unstructured documents:

- The lexical diversity of financial metrics is bounded compared to the open-ended diversity of named entities. There are possibly millions of possible names, but only a limited set of possible financial metrics. On the other hand, for any given metric, a different level of lexical diversity is expected. For instance net income is often expressed as "Net Income", "Net Loss", or "Net Income (Loss)". But current debt can be broken down and expressed in more diverse ways.
- Even though the problem can be thought of as closed-set entity tagging (due to the limited number of metrics), some metrics can be broken down to arbitrarily nuanced sub-metrics. For instance "Selling, General, and Administrative Expenses" may be expressed as "SG&A" or broken down into "Selling and Marketing" and "General and Administrative" items. "Cost of Goods Sold" can be broken down into costs associated with individual products, etc.
- Addition or modification of specific terms can completely change the meaning of a metric. For instance "Revenue" and "Revenue Growth" are different metrics despite lexical overlap, as are "Interest Expense" and "Interest Income". On the other hand, "Net income" and "Net income attributable to XYZ Corporation (GAAP)" might refer to the same metric.
- Positional cues are important to the way the metrics can be perceived. For instance, revenue expressed in the income statement table might refer to a different metric from revenue mentioned in the reconciliation table. These tables are not always labeled properly, so relying on rule-based table-tagging methods might result in a fragile system.

3 PROBLEM STATEMENT

Given a span of text, our goal is to map the span to the closed set of 85 financial metrics defined by S&P standards. The span can be tagged as a metric (e.g. "SG&A") or a sub-metric (e.g. "General and Administrative", which is a sub-metric for SG&A). An 86th tag is added representing the "other" category, which captures when the span doesn't fall under any of the standard metrics. The span can represent a cell in a table, a sentence, or a line-item on a bullet list. The system needs to be implemented in such a way that it is scalable to all financial reports. This means that the system cannot

rely on any structural information within the report (such as iXBRL tags), or semantic cues such as the titles of tables.

4 METHODOLOGY

In order to address the problems laid out in section 2, we employed a hybrid approach to metric tagging. We hypothesized that different metrics would respond better to different tagging approaches in different contexts. Table 1 lists some tagging approaches and examples of metrics that they would be most suitable for.

We used an interpolated scoring mechanism to assign a score to each (m_i, c_j) pair, where m_i represents a metric ($i \in \{1 \cdots 86\}$) and c_j represents a span of text where the metric may have been mentioned ($j \in \{1 \cdots M\}$ where M is the number of spans in the document). The interpolation combines the result of the three models listed in table 1, per below:

$s_{ij} = \alpha * s\_ed_{ij} + \beta * s\_trie_{ij} + \gamma * s\_nbc_{ij}$

where s_{ij} is the score assigned to span c_j for metric m_i, s_ed_{ij} is the edit distance score assigned to it, s_trie_{ij} is the prefix-tree match score assigned to it, and s_nbc_{ij} is the NBC score assigned to it. α, β and γ are tuned on a held-out dataset. The held-out set is also used to select an edit distance measure that works best for each metric. The distance measures are selected from the 21 offered by `textdistance` library [3].

4.1 Data and Labels

We collected 6,738 financial reports published between 2016-01-01 and 2018-12-31. The reports belonged to 300 publicly traded U.S. companies within Technology and Energy sectors, and had been previously manually processed by analysts. The financial metrics identified and extracted by the analysts were stored in a relational database. The challenge was to tie each metric to its proper mention in the corresponding report. To do this, we followed the below steps:

(1) For each metric m_i extract its value v_i from the database. If the number is "too round" (i.e. divisible by 1,000) then ignore it. This reduces the likelihood of mismatches.
(2) Otherwise, look up v_i in the corresponding earnings report. The value might have been scaled, so locate any number that matches $v_i * 10^n$ where $n \in \{-9, -6, -3, 0, 3, 6, 9\}$.
(3) If no matches are located, try to find a window of vertically adjacent table cells whose values add up to $v_i * 10^n$. The window can be as small as 2, or as large as the entire table.
(4) If any number or numbers are found, extract the language surrounding them. If the number has occurred in a table cell, then extract the corresponding non-numeric text from the leftmost cells of row. If the number has occurred in a paragraph, then extract the surrounding sentence [4] surrounding the number. If the number has occurred in a bullet list, then extract the line item including the number. We will refer to this as the context of the metric, or c_j.
(5) Use the (m_i, c_j) pair for training purposes, where the goal is to map each c_j to the proper m_i.

This resulted in 487,722 tuples, where each metric was represented by an average of 5,606 and minimum of 1,731 tuples. We augmented this set by adding 100,000 tuples from spans that had not been mapped to any metric. The dataset was split into a 60% training set, 20% validation set, and 20% test set. The splits were done in a stratified fashion to ensure representation for each metric. The interpolated model was then trained and tuned on the data. For each metric, a threshold was tuned on s_i for baseline rejection. The threshold was tuned to maximize precision for the "other" category. Among all of the metrics that passed the threshold for a given span, the one with the highest score would be selected.

Table 2 breaks down the *F1* performance of each individual model, in addition to the performance of the interpolated model. The F1 measure has been macro-averaged across all metrics.

5 SYSTEM COMPONENTS

Figure 1 illustrates how the data flows through SPread. Each component is described in detail in the following sub-sections.

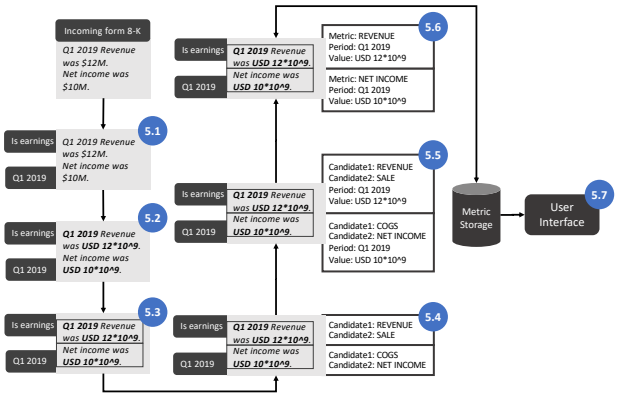

Figure 1: How the documents flow through SPread. Each component has been tagged by the section where it is described in detail.

5.1 Document Ingestion and Classification

SPread ingests forms 8-K as they are posted to the SEC website through a proprietary sourcing engine that uses the RSS services provided by SEC. Forms 8-K are not limited to earnings reports, they can be any press release or general announcement by the company. A taxonomy-based module processes each document and identifies whether earnings language is prevalent in the document. The module assigns a score to the document based on its vocabulary, against a historic dataset created on 231 earnings statements. If the document is determined to be an earnings report, the period that it reports on is inferred based on the below algorithm:

(1) Find the month and day representing the end of fiscal year for the company, as expressed in the document's metadata [5].
(2) Based on the fiscal year information, determine which quarter the report was published in.
(3) Assume that the report covers the previous quarter (because earnings are often released shortly after the quarter is over).

[3] https://github.com/life4/textdistance
[4] We use spaCy for sentence segmentation: https://spacy.io
[5] The SEC includes this information under the `FISCAL-YEAR-END` tag

Table 1: Different methods used for tagging metrics and their advantage.

Method	Description	Suitable for
Edit distance	Calculate an edit distance score between each target metric and each span.	Mapping slightly diverse language such as "Operating Cash Flow" and "Cash Flow form Operations".
Prefix tree	Store metric names in a Trie and match against spans.	Mapping spans such as "Net income attributable to XYZ".
Naive Bayes	Do a one-vs-all NBC [4] for each metric.	Differentiating important terms such as "expense" vs. "income" .

A study on 500 randomly selected 8-K forms (including 242 earnings reports) determined that the classification approach has an accuracy of 86% (erring on the side of a high recall of 98%) and the period inference method works for 92% of the reports.

5.2 Table, Number and Period Normalization

Once a document is determined to be an earnings report, it passes through a few normalization modules. These modules ensure that tables are processed properly. This ensures that merged cells are normalized and hierarchical information is preserved (see figure 2 for an example). This is done using `Selenium for Python` [6], a library that enables us to track the x and y coordinates of each cell in each table and determine their correspondence patterns.

	Quarters Ended		
	June 28, 2019	March 29, 2019	June 29, 2018
Operating expenses:			
Research and development	147.0	151.8	167.1
Selling and marketing	73.6	77.1	81.7
General and administrative	74.1	72.9	74.6

Figure 2: A typical table found in a financial report. The solid blue box highlights rows where left-indentation has been used to indicate hierarchy. The dashed red box show how merged cells can align with multiple columns.

In addition, each number in the document is normalized to a period, a currency value, or other. This is done by a sophisticated system of regular expressions implemented as a wrapper on top of the `datefinder` library [7]. Currency values are scaled based on textual cues (e.g. "$12MM") or the nearest scale indicator (e.g. "All numbers expressed in $thousands").

5.3 Document Segmentation

Next, the document's contents are broken up into segments. Each segment represents a span, i.e. a table cell, a bullet list item, or a sentence. The segments are tagged using unique hash ids. This is done in order to keep track of where each metric is extracted from.

5.4 Metric Tagging

Next, the scoring mechanism described in section 6 is applied to each span. Instead of choosing a single best metric, all of the metrics that pass the s_{ij} threshold are passed on to the next module.

[6] https://selenium-python.readthedocs.io
[7] https://github.com/akoumjian/datefinder

5.5 Mapping to Periods and Values

This module ensures that each candidate is tied to a value (e.g. "USD 12,000,000") and a period (e.g. "Q1 2019"). The process that maps candidates to corresponding periods and values will be disclosed in detail in upcoming publications. If a sub-set of candidates are sub-metrics (e.g. "Selling and Marketing" or "General and Administrative"), they are grouped by corresponding periods. Sub-metric candidates from a given period are summed up to create a new candidate for the given metric (e.g. "SG&A").

5.6 Metric Validation

We apply four validation steps to the candidates.

- First, we compare each candidate's value with historical values for the same metric from the same company. Using MAD analysis [6] we determine whether the new value diverges from historical trends by a cutoff of 2.7 or more. Candidates with diverging values are removed from the pool.
- As expressed in section 2, certain metrics are expected to be expressed in certain tables. Let's call these "sister metrics". If a candidate is located in a table where no sister metrics are present, while other candidates do have sister metrics in the same table as them, then it is removed from the pool.
- Certain metrics need to add up cumulatively (e.g. the sum of revenues from four quarters needs to be equal to the full year revenue), while some metrics are expressed as point-in-time values. If certain candidates violate these conditions, they are removed from the pool.
- Certain metrics are expected to be higher or lower than certain other metrics. For instance net income is expected to be less than revenue. If certain candidates violate these rules (e.g. a candidate for net income has a higher value than a candidate for revenue) then the candidate with the lower s_{ij} score is removed from the pool.

The candidates that survive the validation steps are ranked by their score and the candidate with the highest score is assigned the span.

5.7 User Interface

The selected metrics are displayed on an interactive web UI (see figure 3). By clicking on each cell, the cursor on the right-hand panel jumps to the span where the number was extracted from. The UI allows users to further validate and adjust the values by directly editing the cells. Additionally, historical numbers are displayed for year-on-year and quarter-on-quarter comparison. The figure shows examples of single-line metrics (such as net income) as well as multi-line metrics (selling, general and administrative expenses).

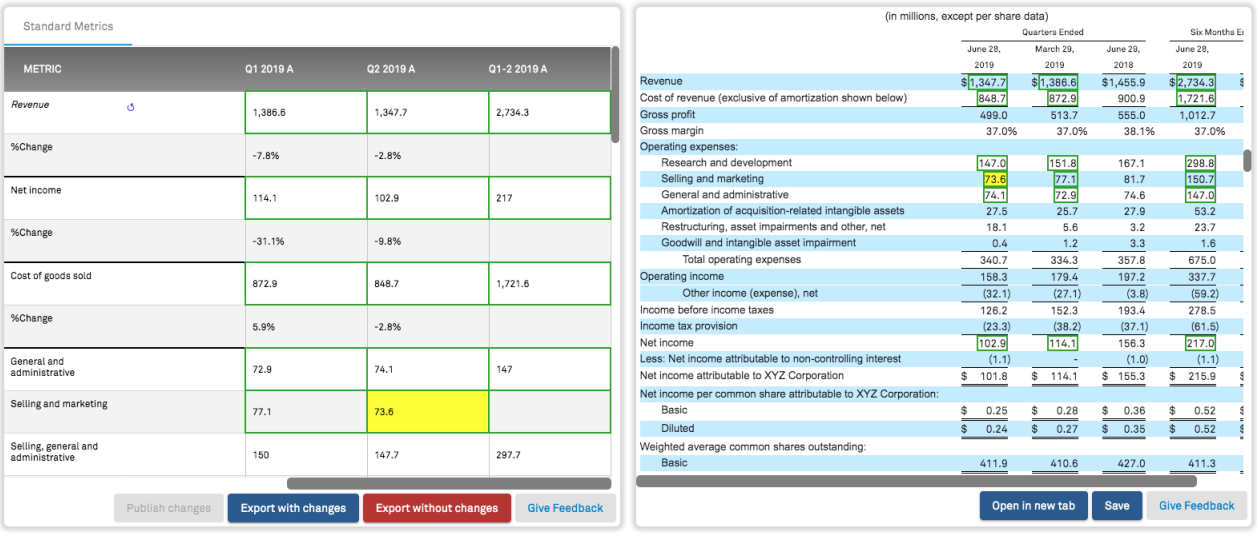

Figure 3: How the results are displayed to the user, linked to the original document, made adjustable and exportable. Note that for this company, the quarter ending on June 28th, 2019 maps to Q1 2019 due to its unique fiscal calendar. "Cost of goods sold" is mapped to "Cost of revenue" as the closest relevant metric. The system recognizes that it needs to break down "Selling, general and administrative" into sub-metrics in order to match the document's disclosure style, while "Revenue" can be expressed as one line item. Clicking into any cell on the left-hand panel will cause it to be highlighted in yellow and the cursor on the right-hand panel jumps to the span where the metric was extracted from (also highlighted in yellow).

Table 2: Macro-averaged F1 scores for each method, broken down by the context where the metrics were expressed in.

Method	Metrics in tables	Metrics in paragraphs
Edit distance	79.2	70.0
Prefix tree	86.1	69.7
NBC	81.3	76.0
Interpolated	85.2	73.2
Validated	**87.3**	**76.6**

6 EVALUATION

Table 2 shows the performance of the system using each individual method outlined in section , as well as the interpolated method and the interpolated+validated method. The results are broken down by whether the metrics were expressed in tables or in paragraphs (bullet lists are also considered non-tabular here). As the table shows, interpolation and validation improve the overall performance of the system, even though paragraph-based metrics temporarily suffer under interpolation but are improved upon validation.

7 CONCLUSION AND FUTURE WORK

In this paper, we described SPread, an automated tool for extracting, spreading and adjusting financial metrics from earning releases in real-time. The tool is able to extract table-based and paragraph-based metrics automatically, and validate its own extractions based on historical trends or expected behavior.

As table 2 shows, the system performs much better on tables than paragraphs. That is due to the structural information that tables provide and the ease of normalizing numbers and scales in this context. In future iterations, we hope to improve the performance of paragraphs by introducing better span segmentation techniques as well as introducing some sequence modeling. Additionally, certain nuances that can impact the performance of the system will be better handled using richer representation of segments. For instance whether Depreciation & Amortization is included or excluded in operating expenses is often disclosed as footnotes surrounding tables. If the document is segmented in such a way that the relevant context surrounding each table is preserved, it will be easier to process and apply these footnotes. Finally, the validation layer provides rich information that can help improve the system's performance in real-time. In addition, as more and more companies switch to the iXBRL format, these tags can be used to tune and improve system performance. This helps bring us closer to the ultimate goal, which is to create an open-ended metric tagging and extraction system.

REFERENCES

[1] [n. d.]. Generally Accepted Accounting Principles (GAAP). https://www.investopedia.com/terms/g/gaap.asp.
[2] [n. d.]. Inline XBRL. https://www.sec.gov/structureddata/osd-inline-xbrl.html.
[3] Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the ACL* 4 (2016), 357–370.
[4] V. P. Deshpande, R. F. Erbacher, and C. Harris. 2007. An Evaluation of NaÃŕve Bayesian Anti-Spam Filtering Techniques. In *2007 IEEE SMC Information Assurance and Security Workshop*. 333–340.
[5] Jenny R. Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on ACL*. 363–370.
[6] Peter J. Rousseeuw and Christophe Croux. 1993. Alternatives to the Median Absolute Deviation. *J. Amer. Statist. Assoc.* 88, 424 (1993), 1273–1283.