Paper SAS1962-2018

Analyzing Text In-Stream and at the Edge

Simran Bagga and Saratendu Sethi, SAS Institute Inc.

ABSTRACT

As companies increasingly use automation for operational intelligence, they are deploying machines to read, and interpret in real time, unstructured data such as news, emails, network logs, and so on. Real-time streaming analytics maximizes data value and enables organizations to act more quickly. For example, being able to analyze unstructured text in-stream and at the “edge” provides a competitive advantage to financial technology (fintech) companies, who use these analyses to drive algorithmic trading strategies. Companies are also applying streaming analytics to provide optimal customer service at the point of interaction, improve operational efficiencies, and analyze themes of chatter about their offerings. This paper explains how you can augment real-time text analytics (such as sentiment analysis, entity extraction, content categorization, and topic detection) with in-stream analytics to derive real-time answers for innovative applications such as quant solutions at capital markets, fake-news detection at online portals, and others.

INTRODUCTION

Text analytics is appropriate when the volume of unstructured text content can no longer be economically reviewed and analyzed manually. The output of text analytics can be applied to a variety of business use cases: detecting and tracking service or quality issues, quantifying customer feedback, assessing risk, improving operational processes, enhancing predictive models, and many more. SAS® Visual Text Analytics provides a unified and flexible framework that enables you to tackle numerous use cases by building a variety of text analytics models. A pipeline-based approach enables you to easily connect relevant nodes that you can use to generate these models.

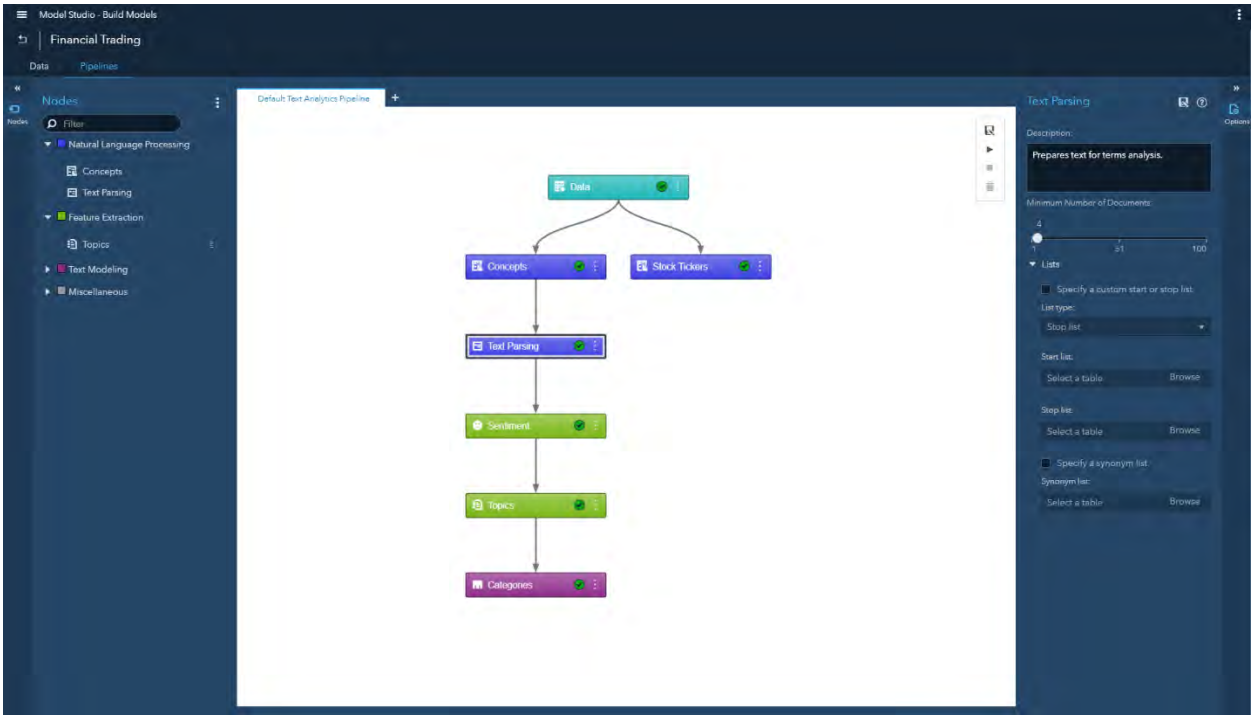
Concepts models enable you to extract entities, concepts, and facts that are relevant to the business. Topic models exploit the power of natural language processing (NLP) and machine learning to discover relevant themes from text. You can use Categories and Sentiment models to tag emotions and reveal insights and issues.

Growing numbers of devices and dependency on Internet of Things (IoT) are causing an increasing need for faster processing, cloud adoption, edge computing, and embedded analytics. The ability to analyze and score unstructured text in real time as events are streaming in is becoming more critical than ever. This paper outlines the use of SAS Visual Text Analytics and SAS® Event Stream Processing to demonstrate a complex event processing scenario. Text models for concept extraction, document categorization, and sentiment analysis are deployed in SAS Event Stream Processing to gain real-time insights and support decision making that is based on intelligence gathered from streaming events.

Big data typically come in dribs and drabs from various sources such as Facebook, Twitter, bank transactions, sensor reading, logs, and so on. The first section of this paper uses SAS Visual Text Analytics to analyze data from trending financial tweets. The latter half focuses on the deployment of text models within SAS Event Stream Processing to assess market impact and intelligently respond to each of the events or data streams as they come in.

EXTRACTING INTELLIGENCE FROM UNSTRUCTURED TEXT USING SAS VISUAL TEXT ANALYTICS

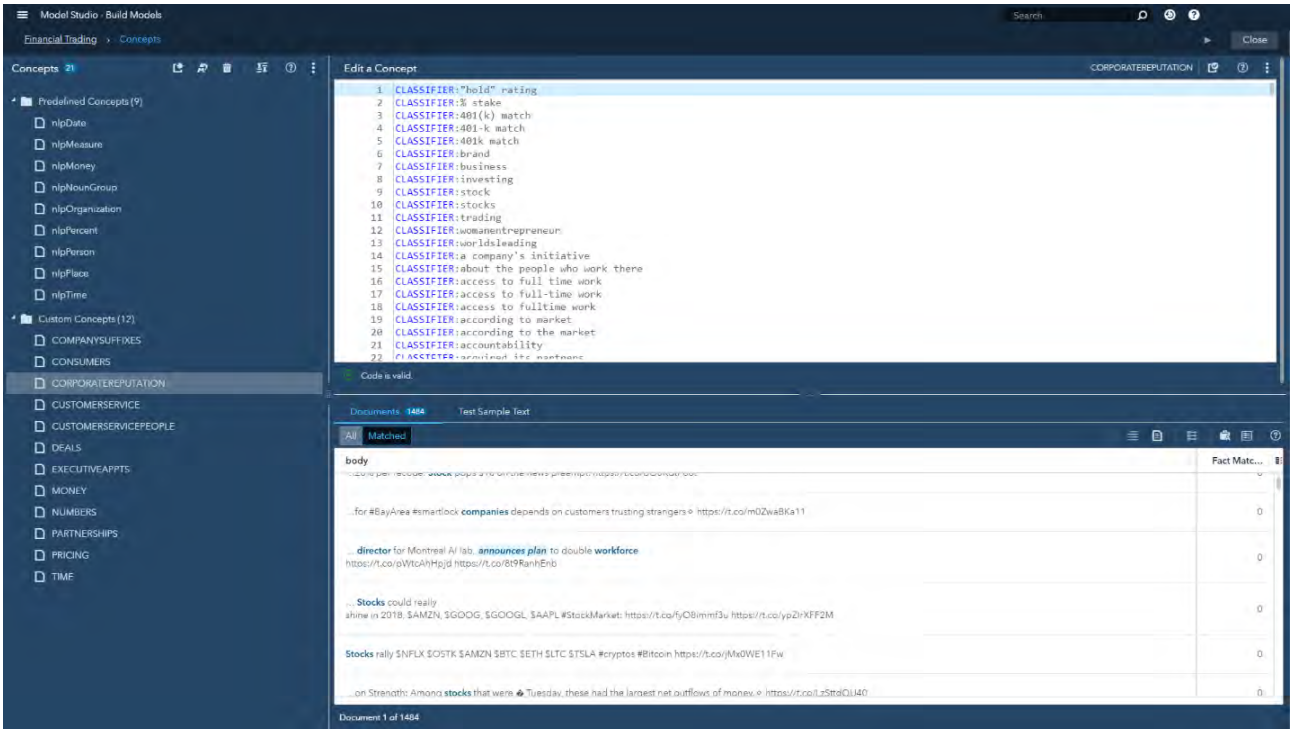
SAS Visual Text Analytics provides a modern, flexible, and end-to-end analytics framework for building a variety of text analytics models that address many use cases. You can exploit the power of natural language processing (NLP), machine learning, and linguistic rules within this single environment. The main focus of NLP is to extract key elements of interest, which can be terms, entities, facts, and so on. Display 1 demonstrates a custom pipeline that you might assemble for a text analytics processing flow. The Concepts node and the Text Parsing node give you the flexibility to enhance the output of NLP and customize the extraction process.



Display 1. Custom Pipeline in SAS Visual Text Analytics

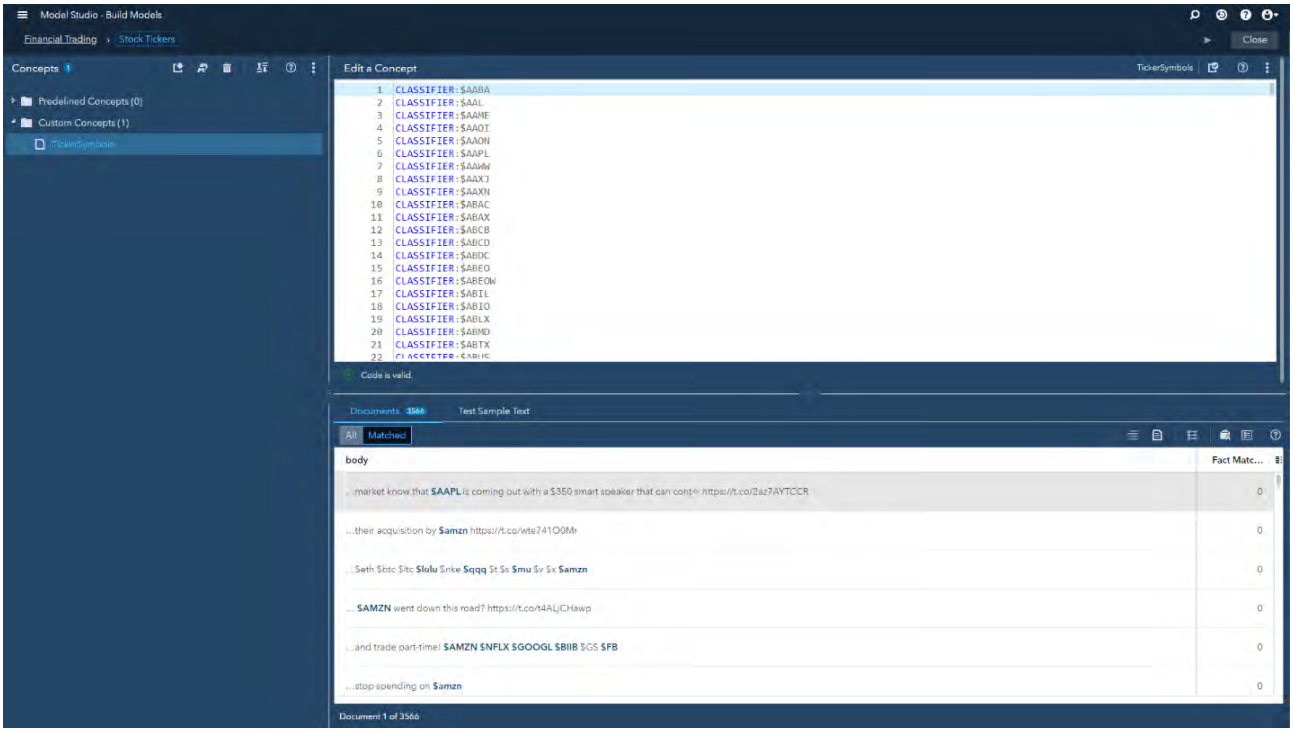
The following list describes the role of each node in this custom pipeline.

- In the Concepts node, you include predefined concepts such as nlpDate, nlpMoney, nlpOrganization, and so on. In this node, you can also create custom concepts and extend the definitions for predefined concepts that are already built into the software. Display 2 shows some custom concepts that have been built to extract information that is related to customer service, corporate reputation, executive appointment, partnerships, and so on, and is likely to affect market trading and volatility. These custom concepts are used for associating categories to each event in SAS Event Stream Processing and will enable automatic concept extraction in future narratives.



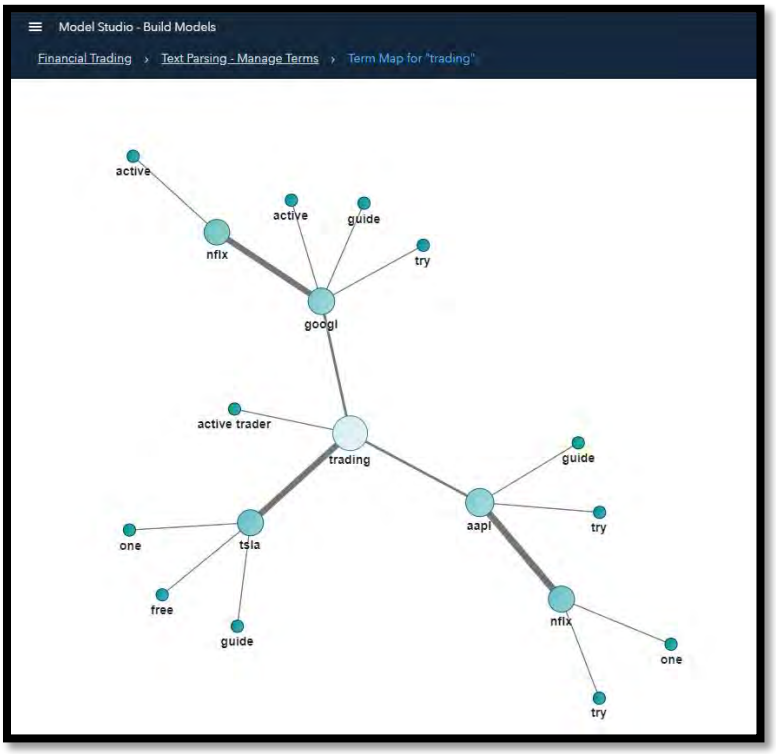
Display 2. Concepts Extraction in SAS Visual Text Analytics

In addition, a custom concepts model is also developed to identify stock ticker symbols in each event. This custom concept model is shown in Display 3.



Display 3. Extracting Stock Ticker Symbols from Text in SAS Visual Text Analytics

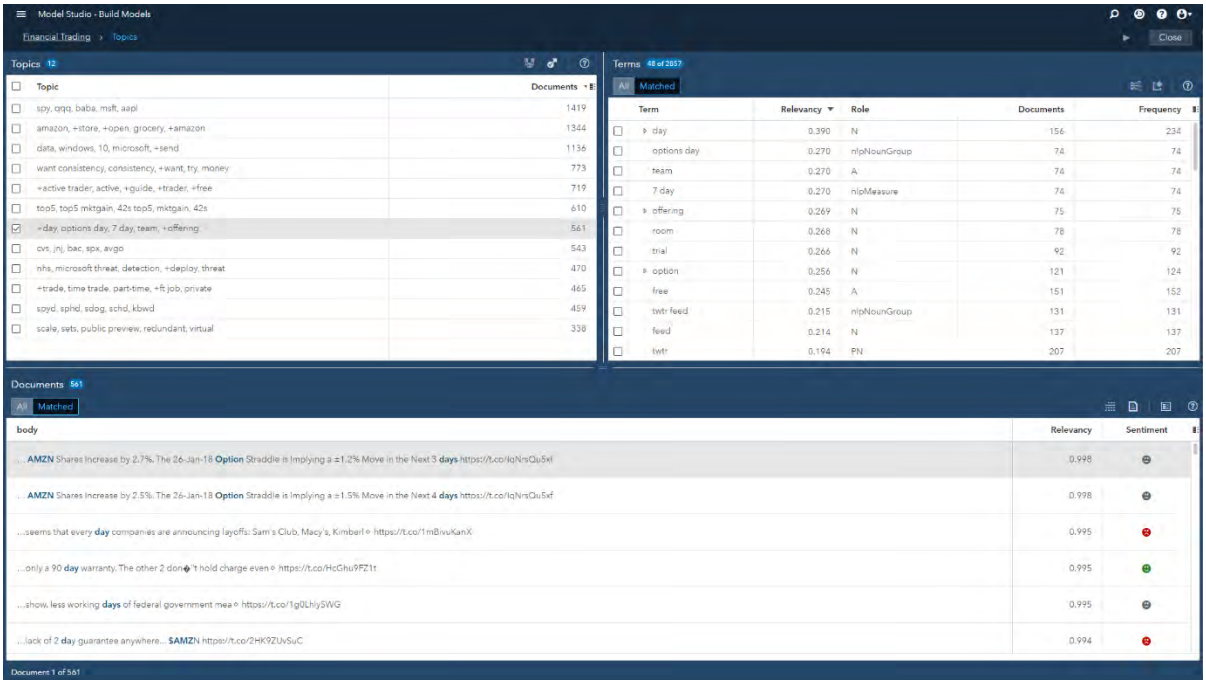
- The Text Parsing node automatically extracts terms and noun groups from text by associating different parts of speech and understanding the context. Recommended lists of Keep and Drop terms are displayed in the interactive window. After the node execution is complete, you can right-click on the node to open the interactive window and drop terms that are not relevant for downstream analysis. The Term Map within the interactive window helps you understand the association of other terms to the term “trading.” See Display 4.



Display 4. Term Map in SAS Visual Text Analytics

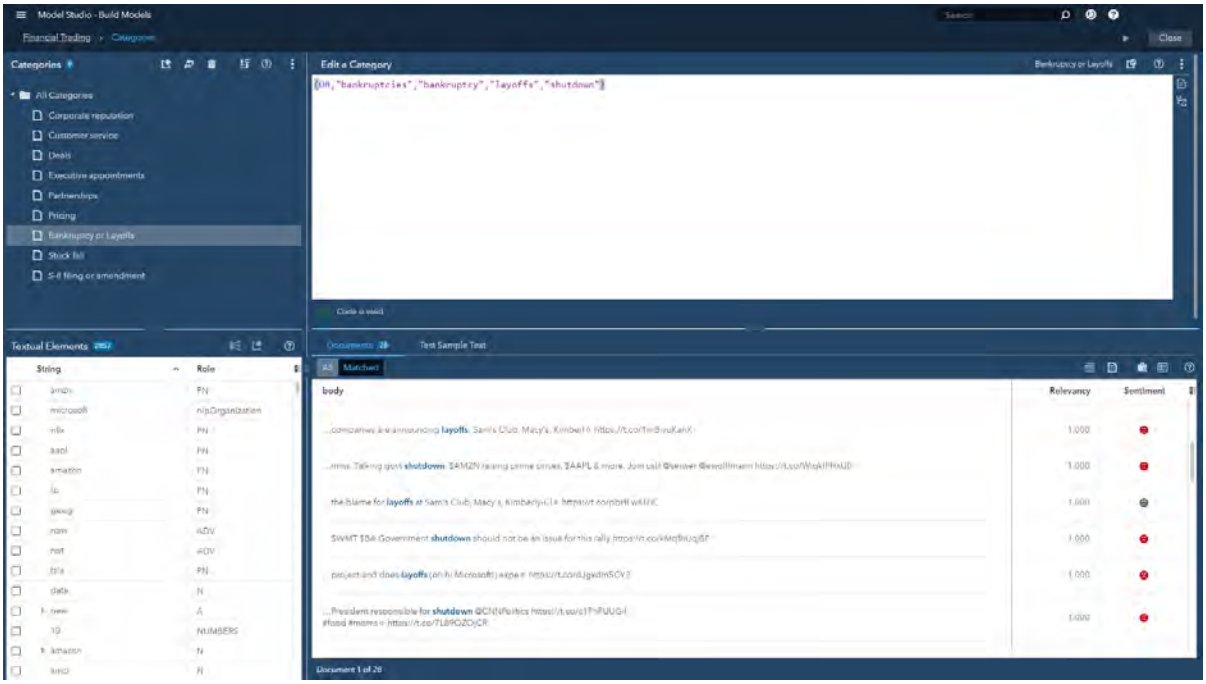
- The Sentiment node uses a domain-independent model that is included with SAS Visual Text Analytics. This rules-based analytic model computes sentiment relevancy for each post and classifies the emotion in unstructured text as positive, negative, or neutral. You can deploy the sentiment model in SAS Event Stream Processing to tag emotions that are associated with a post and that might affect trading decisions.

- The final list of terms from text parsing are fed into machine learning for topic detection. In the interactive window of the Text Topics node (see Display 5), you can see commonly occurring themes within a set of tweets. For example, if you select your topic of interest as “+day, options day, 7 day, team, +offering,” the Documents pane shows all the tweets that mention that topic and the terms that exist within that topic, in addition to relevancy and sentiment. You can deploy the Topics model in-stream in order to capture themes as data or events are streaming in. You can also promote topics of interest into your Categories model, which you can deploy in order to classify text into multiple categories. The implementation of this example uses some categories that were created by promoting relevant topics.



Display 5. Text Topics in SAS Visual Text Analytics

- In the Categories node, you see the taxonomy (Display 6) that has been designed for document categorization. You can manually extend the auto-generated rules from promoted topics and refer to the previously created concepts within your category rules. You can also use the Textual Elements table to select elements of interest that can be inserted into new rules. Multiple posts or tweets about bankruptcy or layoffs, or about an increase or decrease in the number of shares, often result in stock trading shortly thereafter. This intelligence, if available in real time, can aid in buy or sell decisions that are related to that company.



Display 6. Categorization in SAS Visual Text Analytics

SCORING FINANCIAL POSTS IN REAL TIME TO ASSESS MARKET IMPACT

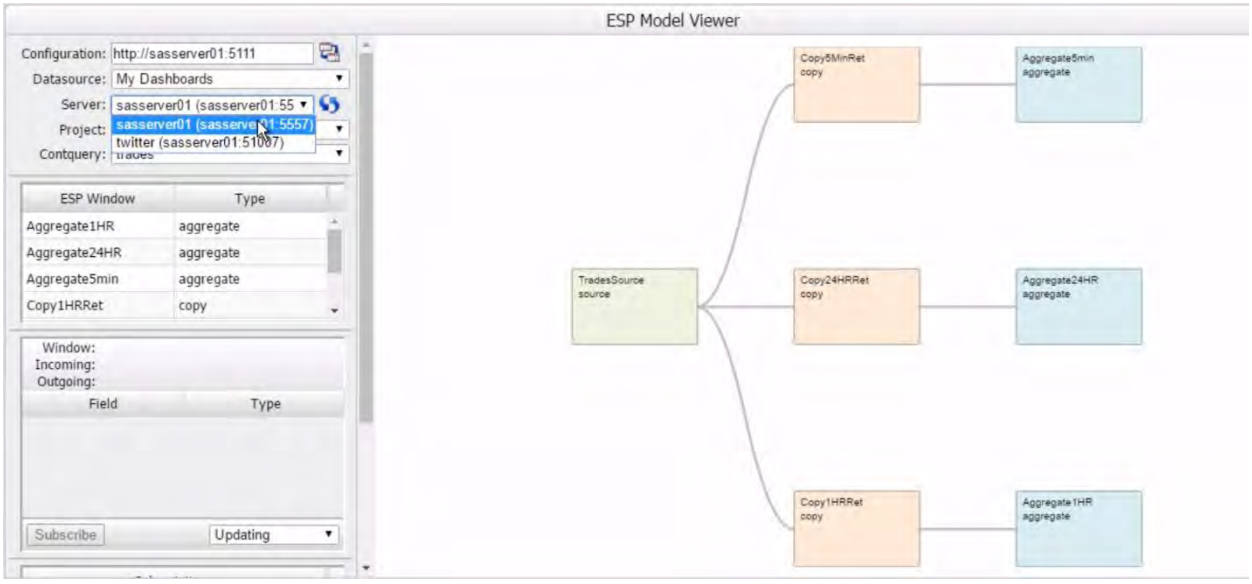
SAS Event Stream Processing is a streaming engine that enables you to analyze or score data as they stream in, rather than first storing them in the database and then analyzing and scoring them in batch. Being able to react to the clicks and events as they are coming in reduces time to action. Event stream processing can occur in three distinct places: at the edge of the network, in the stream, or on data that's at rest (out of the stream).

The SAS Event Stream Processing engine is a transformation engine that augments and adds value to incoming event streams. It is capable of processing millions of events per second. You can perform traditional data management tasks such as filtering out unimportant events, aggregating data, improving data quality, and applying other computations. You can also perform advanced analytic tasks such as pattern detection and text analysis. Events coming in from any source—sensors, Wall Street feeds, router feeds, message buses, server log files—can be read, analyzed, and written back to target applications in real time.

COMPARING STOCK TRADING WEIGHTED AVERAGE PRICE OVER THREE RETENTION PERIODS

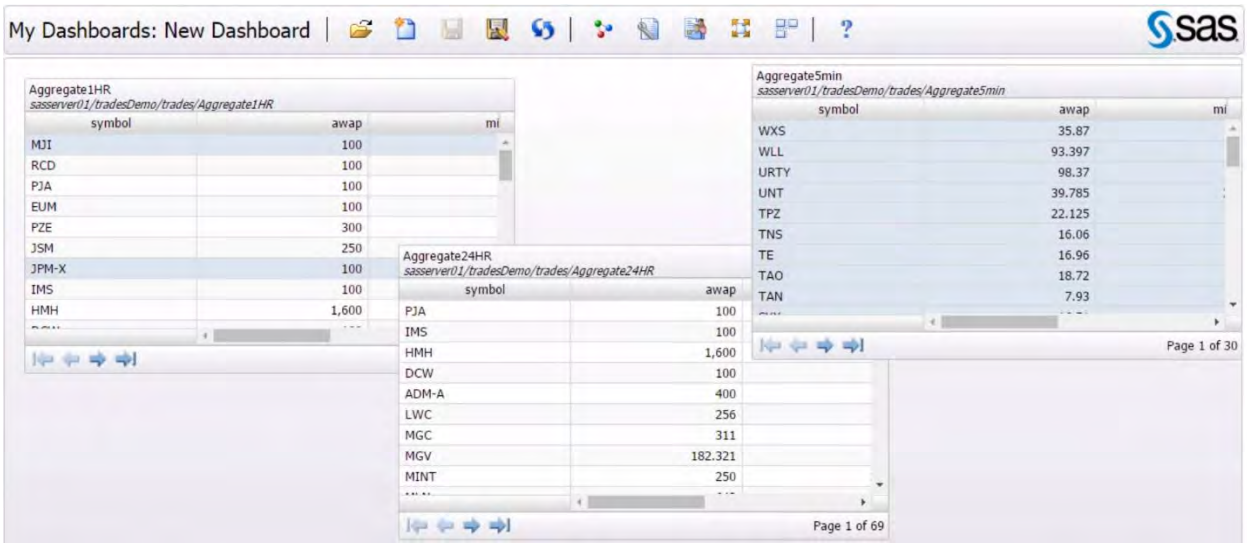
The SAS Event Stream Processing studio is a development and testing application for event stream processing (ESP) models. An ESP model is a program or set of instructions that transforms the input event streams into meaningful output event streams. Once the models are built, they can be published into SAS Event Stream Processing for scoring.

In the ESP model presented in Display 7, the Source window (named TradesSource) is reading from one million stock trades, which are all structured data. The three Copy windows define three different levels of event retention: 5 minutes, 1 hour, and 24 hours. The three Aggregate windows create weighted average trade amounts by stock symbol.



Display 7. Model Viewer in SAS Event Stream Processing

The Stream Viewer window in SAS Event Stream Processing provides a dashboard that enables you to visualize streaming events. This example creates three subscriptions for the three aggregate windows, which can viewed in the dashboard of the Stream Viewer. The dashboard in Display 8 compares the stock trading weighted average price over three retention periods: 5 minute, 1 hour, and 24 hours. The 5-minute view shows what the market is doing right now, whereas the 24-hour view shows what the full day of the market looks like.

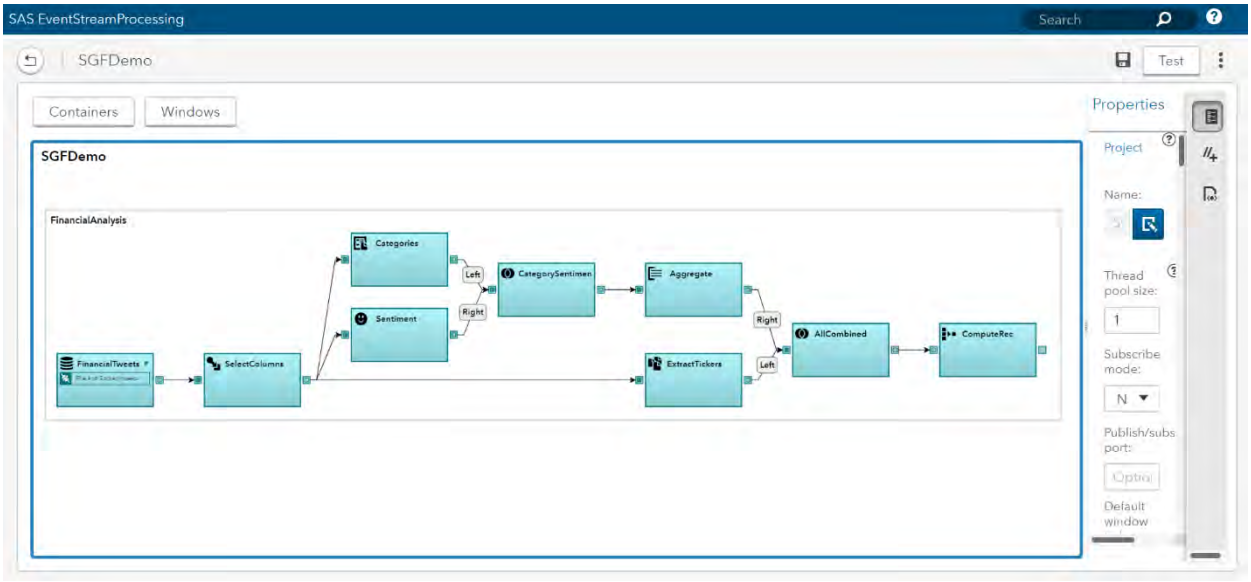


Display 8. Dashboard Viewer in SAS Event Stream Processing

STOCK RECOMMENDATION BASED ON ANALYSIS OF UNSTRUCTURED TEXT

The models that are built using SAS Visual Text Analytics can applied in batch, in-Hadoop, in-stream, and at the edge. This section uses SAS Event Stream Processing to extract concepts, analyze sentiment about particular companies and their stock, and categorize posts as events stream in real time.

In the process defined in Display 9, tweets are continuously flowing through. The Source window (named FinancialTweets) has a retention policy of 15 minutes, which means that the analysis recommendation is based on the last 15 minutes of captured events. As the tweets come in, they are analyzed: stocks tickers are extracted, sentiment score is assigned, and the content is tagged for appropriate categories.



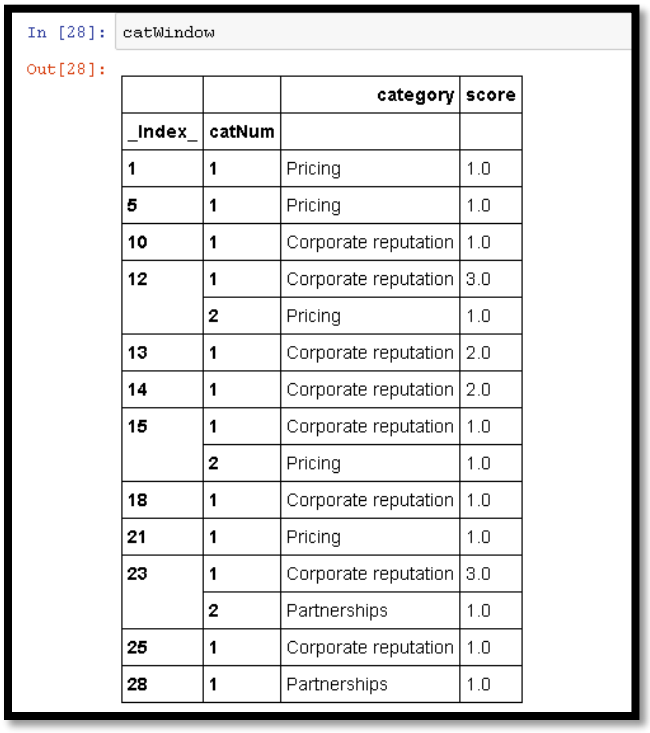
Display 9. SAS Event Stream Processing Studio

The following list describes each window in Display 9 and its role in the flow.

- FinancialTweets: This is a Source window, which is required for each continuous query. All event streams enter continuous queries by being published (injected) into a Source window. Event streams cannot be published into any other window type. Source windows are typically connected to one or more derived windows. Derived windows can detect patterns in the data, transform the data, aggregate the data, analyze the data, or perform computations based on the data. This example uses a CSV (comma-separated values) file with a small sample of tweets that are related to financial and corporate information. Because the sample is small, the results derived here are purely a proof of concept rather than a true financial analysis for all publicly traded companies. For a true streaming use case, SAS Event Stream Processing provides a Twitter adapter, which can be used to feed tweets in real time.

- SelectColumns: This Compute window enables a one-to-one transformation of input events to output events through computational manipulation of the input event stream fields. You can use the Compute window to project input fields from one event to a new event and to augment the new event with fields that result from a calculation. You can change the set of key fields within the Compute window. This example uses the SelectColumns window to filter out attributes that are not relevant for downstream analysis.

- Categories: This is a Text Category window, which categorizes a text field in incoming events. The text field can generate zero or more categories, with scores. Text Category windows are insert-only. This example uses the model file (.mco) that is generated by the **Download Score Code** option of the Categories node in SAS Visual Text Analytics. Display 10 shows the output that is generated by this window. The output lists the document ID (_Index_ column), category number (catNum column), tagged category (category column), and the relevancy score for assigned categorization (score column).



In [28]: catWindow

Out[28]:

		category	score
Index	catNum		
1	1	Pricing	1.0
5	1	Pricing	1.0
10	1	Corporate reputation	1.0
12	1	Corporate reputation	3.0
	2	Pricing	1.0
13	1	Corporate reputation	2.0
14	1	Corporate reputation	2.0
15	1	Corporate reputation	1.0
	2	Pricing	1.0
18	1	Corporate reputation	1.0
21	1	Pricing	1.0
23	1	Corporate reputation	3.0
	2	Partnerships	1.0
25	1	Corporate reputation	1.0
28	1	Partnerships	1.0

Display 10. Text Category Window Output

- Sentiment: This is a Text Sentiment window, which determines the sentiment of text in the specified incoming text field and the probability of its occurrence. The sentiment value is positive, neutral, or negative. The probability is a value between 0 and 1. Text Sentiment windows are insert-only. This example uses the domain-independent sentiment model file (en-base.sam), which is included in SAS Visual Text Analytics. Display 11 shows the output that is generated by this window. Upon scoring, each document in the _Index_ column is assigned an appropriate sentiment tag (in the sentiment column) along with a relevancy score (in the probability column).

```
In [24]: sentWindow
```

Out[24]:

	sentiment	probability
Index		
1	Positive	0.600000
2	Positive	0.600000
3	Neutral	0.500000
4	Neutral	0.500000
5	Neutral	0.500000
6	Neutral	0.500000
7	Neutral	0.500000
8	Neutral	0.500000
9	Neutral	0.500000
10	Positive	0.692308
11	Positive	0.600000
12	Neutral	0.500000
13	Positive	0.600000
14	Positive	0.600000
15	Positive	0.600000

Display 11. Text Sentiment Window Output

- CategorySentiment: This is a Join window, which receives events from an input window to the left of the Join window and produces a single output stream of joined events. Joined events are created according to a user-specified join type and user-defined join conditions. This example does an inner join between the category and sentiment tables to create joined events only when one or more matching events occur on the side opposite the input event. Display 12 shows the output that is generated by the CategorySentiment window.

```
In [25]: catSentWindow
```

Out[25]:

		category	sentiment	probability
Index	catNum			
1	1	Pricing	Positive	0.600000
5	1	Pricing	Neutral	0.500000
10	1	Corporate reputation	Positive	0.692308
12	1	Corporate reputation	Neutral	0.500000
	2	Pricing	Neutral	0.500000
13	1	Corporate reputation	Positive	0.600000
14	1	Corporate reputation	Positive	0.600000
15	2	Pricing	Positive	0.600000
	1	Corporate reputation	Positive	0.600000
18	1	Corporate reputation	Positive	0.600000
21	1	Pricing	Neutral	0.500000
23	2	Partnerships	Neutral	0.500000
	1	Corporate reputation	Neutral	0.500000
25	1	Corporate reputation	Positive	0.600000

Display 12. Joining Category and Sentiment Output Using an Inner Join

- **Aggregate:** Aggregate windows are similar to Compute windows in that non-key fields are computed. Incoming events are placed into aggregate groups such that each event in a group has identical values for the specified key fields. This example aggregates category and sentiment information by stock ticker, as shown in Display 13.

```
In [26]: resWindow
```

Out[26]:

		term	category	sentiment
index	termID			
1	1	$AAPL	Pricing	Positive
5	1	$AMZN	Pricing	Neutral
10	1	$NFLX	Corporate reputation	Positive
	2	$AMZN	Corporate reputation	Positive
12	1	$AMZN	Corporate reputation|Pricing	Neutral
15	3	$AMZN	Pricing|Corporate reputation	Positive
	1	$AAPL	Pricing|Corporate reputation	Positive
	2	$QQQ	Pricing|Corporate reputation	Positive
18	3	$GOOGL	Corporate reputation	Positive
	2	$GOOG	Corporate reputation	Positive
	1	$AMZN	Corporate reputation	Positive
	4	$AAPL	Corporate reputation	Positive
21	2	$AMZN	Pricing	Neutral
	1	$AMZN	Pricing	Neutral

Display 13. Joining Category and Sentiment Output with Extracted Ticker Concepts and Aggregating Categories for Each Stock Ticker

- ExtractTickers: This is a Text Context window, which is used here to call the SAS Visual Text Analytics Concepts model to extract key terms or entities of interest from text. Events that are generated from the terms can be analyzed by other window types. For example, a Pattern window could follow a Text Context window to look for tweet patterns of interest. This example combines the extracted tickers with category and sentiment information from posts.

  The stock tickers are extracted by using the model file (.li) that is generated by the **Download Score Code** option of the Concepts node in SAS Visual Text Analytics. This file is also shown in Display 3.

- AllCombined: This is a second Join window; it combines output from the CategorySentiment window with output from the ExtractTickers window. Display 13 shows the output that is generated by this window. In the AllCombined output, categories and sentiment are aggregated across each stock ticker symbol within a particular document. For example, in document ID 15, $AMZN refers to both “Pricing” and “Corporate reputation” categories, with the overall sentiment being positive.

- ComputeRec: This is a Procedural window, which is a specialized window that enables you to apply external methods to event streams. You can use it when complex procedural logic is required or when external methods or functions already exist. You can apply external methods by using C++, SAS DS2, SAS DATA step, or SAS® Micro Analytic Services. This example calls Python through SAS Micro Analytic Services; the code implements custom logic such as the following:

- If sentiment is “Negative” and relevancy is close to 1, then recommend a sell.
- If category is “Executive appointments” and sentiment is “Positive,” then recommend a buy.
- If category is “Corporate reputation” and sentiment is “Positive,” then recommend a hold.

As events continuously flow into the system, a recommendation is assigned for each event. If the post is associated with negative sentiment, then the recommendation would be to sell the stock. Display 14 shows the output and recommendations that are generated by this window.

```
In [27]: procWindow
```

Out[27]:

		recommendation
term	_index_	
$AAPL	1	HOLD
$AMZN	5	HOLD
$NFLX	10	BUY
$AMZN	10	BUY
	12	HOLD
	15	HOLD
$AAPL	15	HOLD
$QQQ	15	HOLD
$GOOGL	18	BUY
$GOOG	18	BUY
$AMZN	18	BUY
$AAPL	18	BUY

Display 14. Procedural Window Showing Final Recommendation for Each Event

OTHER APPLICATIONS

You can also use SAS Visual Text Analytics and SAS Event Stream Processing to address more mature business use cases, such as the following:

- Financial scenarios
  - Quantitative investment and trading strategies: The trading and investment signals from real-time text analytics are applicable across all trading frequencies and provide an incremental source of quantitative factors.
  - Algorithmic trading: You can enhance algorithmic strategies with automated circuit breakers, or you can develop new algorithms that take advantage of the ability to better predict trading volumes, price volatility, and directional movements.
  - Market making: You can widen spreads or pull quotes when significant negative news is affecting a particular stock.
  - Portfolio management: You can improve asset allocation decisions by benchmarking portfolio sentiment.
  - Fundamental analysis: You can forecast stock, sector, and market outlooks.
- Non-financial scenarios
  - Online email analysis of the mail exchange server to detect intellectual property (IP) leakage as emails are coming inbound and going outbound
  - Fake-news detection and its possible impact on the stock market. Fake news can be identified in various ways: by examining the source, its popularity, and trustworthiness (Waldrop 2017).

CONCLUSION

This paper highlights how unstructured text analysis can be applied in-stream to provide a competitive advantage to financial technology institutions that use the analysis to drive algorithmic trading strategies. Although fintechs use more sophisticated algorithms, this approach demonstrates a simplified implementation that is very feasible within the framework of SAS Visual Text Analytics and SAS Event Stream Processing. This paper does not combine the results of structured data and unstructured text from tweets because access to real-time streaming sources was not available.

In-stream analytics occur as data streams from one device to another, or from multiple sensors to an aggregation point. Event stream processing is also supported at the "edge" so that you can analyze any data that are processed on the same device from which they are streaming.

REFERENCE

- Waldrop, M. Mitchell. 2017. "News Feature: The Genuine Problem of Fake News." *Proceedings of the National Academy of Sciences of the United States of America* 114 (48): 12631–12634. http://www.pnas.org/content/114/48/12631

ACKNOWLEDGMENTS

The authors thank Kevin Smith for helping implement the workflow using a Python interface to SAS Event Stream Processing.

They also thank Anne Baxter for editing the paper.

RECOMMENDED READING

- *SAS® Visual Text Analytics: Programming Guide*
- *SAS® Event Stream Processing: Programming Reference*
- Robert, Nicholas. "How to perform real time text analytics on Twitter streaming data in SAS ESP." Available https://blogs.sas.com/content/sgf/2016/10/05/how-to-perform-real-time-text-analytics-on-twitter-streaming-data-in-sas-esp/. Last modified October 5, 2016. Accessed on February 26, 2018.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Simran Bagga
SAS Institute Inc.
simran.bagga@sas.com

Saratendu Sethi
SAS Institute Inc.
saratendu.sethi@sas.com