Social media analysis: Real-time Twitter sentiment analysis in Azure Stream Analytics

By jeffstokes72
Last updated: 04/28/2015

In this article:

In this tutorial, you'll learn how to build a sentiment analysis solution by bringing real-time Twitter events into Event Hubs, writing Stream Analytics queries to analyze the data, and then storing the results or using a dashboard to provide insights in real time.

Social media analytics tools help organizations understand trending topics, meaning subjects and attitudes with a high volume of posts in social media. Sentiment analysis - also called "opinion mining" - uses social media analytics tools to determine attitudes toward a product, idea, and so on.

Scenario

A news media website is interested in getting an edge over its competitors by featuring site content that is immediately relevant to its readers. They use social media analysis on

topics relevant to their readers by doing real time sentiment analysis on Twitter data. Specifically, to identify what topics are trending in real time on Twitter, they need real-time analytics about the tweet volume and sentiment for key topics.

Prerequisites

1. A Twitter account is required for this tutorial.
2. This walkthough uses an event generator located on GitHub. Download it here and follow the steps below to set up your solution.

Create an Event Hub input and a Consumer Group

The sample application will generate events and push them to an Event Hubs instance (an Event Hub, for short). Service Bus Event Hubs are the preferred method of event ingestion for Stream Analytics. See Event Hubs documentation in Service Bus documentation

Follow the steps below to create an Event Hub.

1. In the Azure portal click **NEW** > **APP SERVICES** > **SERVICE BUS** > **EVENT HUB** > **QUICK CREATE** and provide a name, region, and new or existing namespace to create a new Event Hub.

2. As a best practice, each Stream Analytics job should read from a single Event Hubs Consumer Group. We will walk you through the process of creating a Consumer Group below and you can learn more about them here. To create a Consumer Group, navigate to the newly created Event Hub and click the **CONSUMER GROUPS** tab, then click **CREATE** on the bottom of the page and provide a name for your Consumer Group.

3. To grant access to the Event Hub, we will need to create a shared access policy. Click the **CONFIGURE** tab of your Event Hub.

4. Under **SHARED ACCESS POLICIES**, create a new policy with **MANAGE** permissions.

shared access policies	
NAME	**PERMISSIONS**
manage	Manage, Send, Listen
NEW POLICY NAME	∨

5. Click **SAVE** at the bottom of the page.

6. Navigate to the **DASHBOARD** and click **CONNECTION INFORMATION** at the bottom of the page and copy and save the connection information. (Use the copy icon that appears under the search icon.)

Configure and start event-generator application

We have provided a client application that will tap into Twitter data via Twitter's REST APIs to collect Tweet events about a parameterized set of topics. The 3rd party open source tool Sentiment140 is used to assign a sentiment value to each tweet (0: negative, 2: neutral, 4: positive) and then Tweet events are pushed to Event Hub.

Follow these steps to set up the application:

1. Download the TwitterClient solution

2. Open App.config and replace oauth_consumer_key, oauth_consumer_secret, oauth_token, oauth_token_secret with Twitter tokens with your values.

   Steps to generate an OAuth access token

   Note that you will need to make an empty application to generate a token.

3. Replace the EventHubConnectionString and EventHubName values in App.config with your Event Hub connection string and name.

4. *Optional:* Adjust the keywords to search for. As a default, this application looks for "Azure,Skype,XBox,Microsoft,Seattle". You can adjust the values for twitter_keywords in App.config, if desired.

5. Build the solution

6. Start the application. You will see Tweet events with the CreatedAt, Topic, and SentimentScore values being sent to your Event Hub:

Create Stream Analytics job

Now that we have Tweet events streaming in real-time from Twitter, we can set up a Stream Analytics job to analyze these events in real time.

Provision a Stream Analytics job

1. In the Azure portal, click **NEW** > **DATA SERVICES** > **STREAM ANALYTICS** > **QUICK CREATE**.

2. Specify the following values, and then click **CREATE STREAM ANALYTICS JOB**:

   ○ **JOB NAME**: Enter a job name.

   ○ **REGION**: Select the region where you want to run the job. Consider placing the job and the event hub in the same region to ensure better performance and to ensure that you will not be paying to transfer data between regions.

   ○ **STORAGE ACCOUNT**: Choose the Storage account that you would like to use to store monitoring data for all Stream Analytics jobs running within this region. You

have the option to choose an existing Storage account or to create a new one.

3. Click **STREAM ANALYTICS** in the left pane to list the Stream Analytics jobs.

4. The new job will be shown with a status of **CREATED**. Notice that the **START** button on the bottom of the page is disabled. You must configure the job input, output, and query before you can start the job.

Specify job input

1. In your Stream Analytics job click **INPUTS** from the top of the page, and then click **ADD INPUT**. The dialog that opens will walk you through a number of steps to set up your input.

2. Select **DATA STREAM**, and then click the right button.

3. Select **EVENT HUB**, and then click the right button.

4. Type or select the following values on the third page:

   o **INPUT ALIAS**: Enter a friendly name for this job input, such as TwitterStream. Note that you will be using this name in the query later on. **EVENT HUB**: If the Event Hub you created is in the same subscription as the Stream Analytics job, select the namespace that the event hub is in.

   If your event hub is in a different subscription, select **Use Event Hub from Another Subscription**, and then manually enter information for **SERVICE BUS NAMESPACE**, **EVENT HUB NAME**, **EVENT HUB POLICY NAME**, **EVENT HUB POLICY KEY**, and **EVENT HUB PARTITION COUNT**.

   o **EVENT HUB NAME**: Select the name of the Event Hub

   o **EVENT HUB POLICY NAME**: Select the event-hub policy created earlier in this tutorial.

   o **EVENT HUB CONSUMER GROUP**: Type in the Consumer Group created earlier in this tutorial.

5. Click the right button.

6. Specify the following values:

   o **EVENT SERIALIZER FORMAT**: JSON

   o **ENCODING**: UTF8

7. Click the check button to add this source and to verify that Stream Analytics can successfully connect to the event hub.

Specify job query

Stream Analytics supports a simple, declarative query model for describing transformations. To learn more about the language, see the Azure Stream Analytics Query Language Reference. This tutorial will help you author and test several queries over Twitter data.

Sample data input

To validate your query against actual job data, you can use the SAMPLE DATA feature to extract events from your stream and create a .JSON file of the events for testing.

1. Select on your Event Hub input and click **SAMPLE DATA** at the bottom of the page.

2. In the dialog that appears, specify a **START TIME** to start collecting data from and a **DURATION** for how much additional data to consume.

3. Click the **DETAILS** button, and then the **Click here** link to download and save the .JSON file that is generated.

Pass-through query

To start with, we will do a simple pass-through query that projects all the fields in an event.

1. Click **QUERY** from the top of the Stream Analytics job page.

2. In the code editor, replace the initial query template with the following:

```
SELECT * FROM TwitterStream
```

   Make sure that the name of the input source matches the name of the input you specified earlier.

3. Click **TEST** under the query editor

4. Browse to your sample .JSON file

5. Click the check button and see the results displayed below the query definition.

▲ Output

CREATEDAT	TOPIC	SENTIMENTSCORE	EVENTPROCESSEDUTCTIME	PARTITIONID	EVENTENQUEUEDUTCTIME
2015-04-14T04:27:13	Unknown	2	2015-04-14T04:33:10.88046...	3	2015-04-14T04:27:14.874Z
2015-04-14T04:27:16	Microsoft	2	2015-04-14T04:33:09.43090...	0	2015-04-14T04:27:17.265Z
2015-04-14T04:27:19	Unknown	2	2015-04-14T04:33:10.91169...	3	2015-04-14T04:27:19.796Z
2015-04-14T04:27:19	Microsoft	2	2015-04-14T04:33:09.43090...	0	2015-04-14T04:27:20.093Z
2015-04-14T04:27:19	Microsoft	2	2015-04-14T04:33:09.43090...	0	2015-04-14T04:27:20.421Z
2015-04-14T04:27:21	Unknown	4	2015-04-14T04:33:10.95857...	3	2015-04-14T04:27:21.28Z

Count of tweets by topic: Tumbling window with aggregation

To compare the number of mentions between topics, we'll leverage a TumblingWindow to

get the count of mentions by topic every 5 seconds.

1. Change the query in the code editor to:

```
SELECT System.Timestamp as Time, Topic, COUNT(*)
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY TUMBLINGWINDOW(s, 5), Topic
```

Note that this query uses the **TIMESTAMP BY** keyword to specify a timestamp field in the payload to be used in the temporal computation. If this field wasn't specified, the windowing operation would be performed using the time each event arrived at Event Hub. Learn more under "Arrival Time Vs Application Time" in the Stream Analytics Query Reference.

This query also accesses a timestamp for the end of each window with **System.Timestamp**.

2. Click **RERUN** under the query editor to see the results of the query.

Identifying trending topics: Sliding window

To identify trending topics we'll look for topics that cross a threshold value for mentions in a given amount of time. For the purposes of this tutorial, we'll check for topics that are mentioned more than 20 times in 5 seconds using a SlidingWindow.

1. Change the query in the code editor to:

```
SELECT System.Timestamp as Time, Topic, COUNT(*) as Mentions
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY SLIDINGWINDOW(s, 5), topic
HAVING COUNT(*) > 20
```

2. Click **RERUN** under the query editor to see the results of the query.

◢ Output

TIME	TOPIC	MENTIONS
2015-04-14T11:34:08.000Z	Microsoft	22
2015-04-14T12:00:37.000Z	Microsoft	24

Count of mentions and sentiment: Tumbling window with aggregation

The final query we will test uses a TumblingWindow to obtain the number of mentions and average, minimum, maximum, and standard deviation of sentiment score for each topic every 5 seconds.

1. Change the query in the code editor to:

```
SELECT System.Timestamp as Time, Topic, COUNT(*), AVG(SentimentScore), MIN(SentimentScore),
Max(SentimentScore), STDEV(SentimentScore)
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY TUMBLINGWINDOW(s, 5), Topic
```

2. Click **RERUN** under the query editor to see the results of the query.

3. This is the query we will use for our dashboard. Click **SAVE** at the bottom of the page.

Create output sink

Now that we have defined an event stream, an Event Hub input to ingest events, and a query to perform a transformation over the stream, the last step is to define an output sink for the job. We'll write the aggregated tweet events from our job query to an Azure Blob. You could also push your results to SQL Database, Table Store or Event Hub, depending on your application needs.

Follow the steps below to create a container for Blob storage, if you don't already have one:

1. Use an existing Storage account or create a new Storage account by clicking **NEW** > **DATA SERVICES** > **STORAGE** > **QUICK CREATE** > and following the instructions on the screen.

2. Select the Storage account and then click **CONTAINERS** at the top of the page, and then click **ADD**.

3. Specify a **NAME** for your container and set its **ACCESS** to Public Blob.

Specify job output

1. In your Stream Analytics job, click **OUTPUT** at the top of the page, and then click **ADD OUTPUT**. The dialog that opens will walk you through a number of steps to set up your output.

2. Select **BLOB STORAGE**, and then click the right button.

3. Type or select the following values on the third page:

   o **OUTPUT ALIAS**: Enter a friendly name for this job output

   o **SUBSCRIPTION**: If the Blob Storage you created is in the same subscription as the Stream Analytics job, select **Use Storage Account from Current Subscription**. If your storage is in a different subscription, select **Use Storage Account from Another Subscription** and manually enter information for **STORAGE ACCOUNT**, **STORAGE ACCOUNT KEY**, **CONTAINER**.

   o **STORAGE ACCOUNT**: Select the name of the Storage Account

   o **CONTAINER**: Select the name of the Container

   o **FILENAME PREFIX**: Type in a file prefix to use when writing blob output

4. Click the right button.

5. Specify the following values:

   - **EVENT SERIALIZER FORMAT**: JSON

   - **ENCODING**: UTF8

6. Click the check button to add this source and to verify that Stream Analytics can successfully connect to the storage account.

# Start job

Since a job input, query and output have all been specified, we are ready to start the Stream Analytics job.

1. From the job **DASHBOARD**, click **START** at the bottom of the page.

2. In the dialog that appears, select **JOB START TIME**, and then click the checkmark button on the bottom of the dialog. The job status will change to **Starting** and will shortly move to **Running**.

View output for sentiment analysis

Once your job is running and processing the real-time Twitter stream, choose how you want to view the output for sentiment analysis. Use a tool like Azure Storage Explorer or Azure Explorer to view your job output in real time. From here, you could extend your application to include a customized dashboard over your output, like the one pictured below using Power BI.
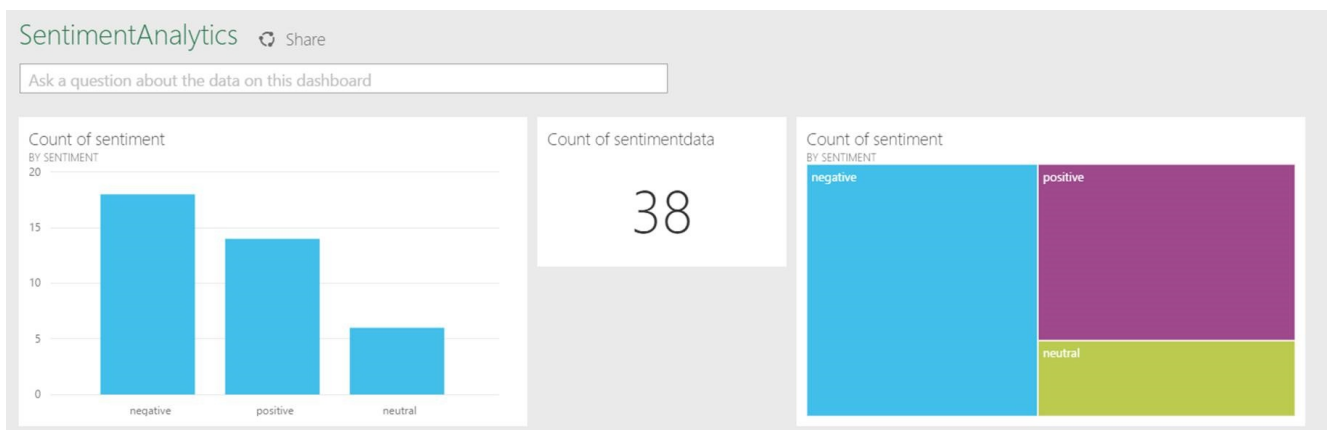
Get support

For further assistance, try our Azure Stream Analytics forum.

Next steps

- Introduction to Azure Stream Analytics
- Get started using Azure Stream Analytics

- [Scale Azure Stream Analytics jobs](#)
- [Azure Stream Analytics Query Language Reference](#)
- [Azure Stream Analytics Management REST API Reference](#)

Need help?

Go to an MSDN forum ▶

StackOverflow discussion ▶

---

17 Comments   **Azure Site**   ❶ **Login** ▾

♥ Recommend   ↱ Share   Sort by Newest ▾

	Join the discussion…

Luis C. Leon · 14 days ago

Hi:

This is such a nice tutorial however I'm getting an exception when calling the method: _eventHubClient.Send(data);

<message>A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond</message>

Exception rethrown at [0]:
at Microsoft.ServiceBus.Common.ExceptionDispatcher.Throw(Exception exception)
at Microsoft.ServiceBus.Common.AsyncResult.End[TAsyncResult](IAsyncResult result)
at Microsoft.ServiceBus.Common.AsyncResult`1.End(IAsyncResult asyncResult)
at Microsoft.ServiceBus.Messaging.Amqp.AmqpMessagingFactory.EndOpenEntity(IAsyn result)
at Microsoft.ServiceBus.Messaging.Amqp.AmqpMessageSender.EndCreateLink(IAsyncF

see more

⌃ | ⌄ · Reply · Share ›

Janet Yeilding [janety@MSFT] ↱ Luis C. Leon · 14 days ago

Janet Yeilding [janety@MSFT] → Luis C. Leon • 14 days ago

Hi Luis, for help debugging this problem please post in the ASA forum: https://social.msdn.microsoft....

∧ | ∨ • Reply • Share ›

Luis C. Leon → Janet Yeilding [janety@MSFT] • 13 days ago

Hi Janet,

Thanks for your reply. I already post in the forum. The link for the question is: https://social.msdn.microsoft....

I appreciate your help

∧ | ∨ • Reply • Share ›

Gene Furibondo • 2 months ago

Any thoughts on how to include lat/long geotag information in this analysis.

∧ | ∨ • Reply • Share ›

artybala • 2 months ago

Hi:

I am getting "The remote server returned an error: (401) Unauthorized" error when i tried run the code. anyone had this issue?

∧ | ∨ • Reply • Share ›

Ziv Kasperski [zivk@MSFT] → artybala • 2 months ago

It might be easier to investigate if you post this question on the Stream Analytics forum here: https://social.msdn.microsoft....
Please upload an image of the error so we have better context to help you.
Thanks!
Ziv.

∧ | ∨ • Reply • Share ›

artybala → Ziv Kasperski [zivk@MSFT] • 2 months ago

Thanks Ziv. posting now

∧ | ∨ • Reply • Share ›

Marian Dragt • 2 months ago

Nice article! However, I have the same comment as Frederik: the JSON array blob misses a]. Any solution?

∧ | ∨ • Reply • Share ›

Burak Kaan Köse • 2 months ago

I've been trying to use Storage Table as an output but couldn't get it work.It says 'Additional details are temporarily unavailable.' everytime ...

Stream Analytics is one of the most powerfull services of Azure but since it's still in preview it lacks a lot of debugging.Is there any other debugging options other than management portal or logs ?

∧ | ∨ • Reply • Share ›

Ziv Kasperski [zivk@MSFT] ➜ Burak Kaan Köse · 2 months ago

Hi Burak,
Did you try to look in the Operations Log?
You can reach them via the quick glance section in the dashboard of your Stream Analytics job.
When I had issues with output to Table Storage it was usually a result of my selected Partition Key column or Row Key column having invalid characters.
Thanks!

Ziv.

∧ │ ∨ · Reply · Share ›

Frederik Willaert · 2 months ago

nice article, fun to play
trying to get some visualization in PowerBI, I find that the json array blob written by Stream Analytics doesn't contain a closing square bracket] at the end which makes PowerBI hickup

∧ │ ∨ · Reply · Share ›

Marian Dragt ➜ Frederik Willaert · 2 months ago

Hi Frederik, I changed the output to csv and then you can connect to it via PowerBI

∧ │ ∨ · Reply · Share ›

Ziv Kasperski [zivk@MSFT] ➜ Marian Dragt · 2 months ago

Hi Guys,

Can you elaborate on how you output to Power BI?
The output to Power BI that is built into Stream Analytics does not take a serialization format at all.
Thanks!
Ziv.

∧ │ ∨ · Reply · Share ›

Marian Dragt ➜ Ziv Kasperski [zivk@MSFT] · 2 months ago

Hi Ziv, if you go to the OUTPUTS tab of your job, and change the EVENT SERIALIZATION FORMAT to csv with comma as delimiter, you will be able to use the data with Power Query

∧ │ ∨ · Reply · Share ›

Ziv Kasperski [zivk@MSFT] ➜ Marian Dragt
· 2 months ago

Hi Marian,

I take it that you are using blob output and have built an Excel workbook with Power Query reading from the blob and you built a Power BI dashboard on top of the Excel workbook, is that correct?

Did you try picking Power BI output (in preview) directly from the add output wizard of Stream Analytics?

Thanks!

Ziv.

see more

⌃ | ⌄ · Reply · Share ›

Marian Dragt → Ziv Kasperski [zivk@MSFT]
· 2 months ago

Hi Ziv, I didn't get that option when creating the output....could you please explain me how to set it up like Power BI output?

⌃ | ⌄ · Reply · Share ›

Ziv Kasperski [zivk@MSFT] → Marian Dragt
· 2 months ago

To get the Power BI output option you need to sign into the Azure Management Portal using an organizational account - more about organizational accounts can be found here: https://msdn.microsoft.com/en-...
You will have to use the same organizational account on both Azure Management Portal and Power BI.
Thanks!
Ziv.

see more

⌃ | ⌄ · Reply · Share ›

Go Social

Microsoft Azure	+

Community	+

Support	+

Account	+

Trust Center	+

Hello from Seattle.

English (US) ⌄

US Dollar ($) ⌄

Contact Us   Feedback   Trademarks   Privacy & Cookies

Microsoft