Sign in (https://twitter.com/login?redirect_after_login=https://developer.twitter.com/en/docs/tutorials/building-high-quality-filters)
Sign up (https://developer.twitter.com/en/portal/petition/essential/basic-info)

Tutorials / Building high-quality filters for getting Twitter data

Building high-quality filters for getting Twitter data

Relevant products:

› Recent search (https://developer.twitter.com/en/docs/twitter-api/tweets/search/introduction)

› Filtered stream (https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream/introduction)

Introduction

With the Twitter API, you can tap into the public conversation to understand what's happening, discover insights, identify trends, and more. The high volume of data and variety in public conversation are big advantages to building your dataset with the Twitter API. By the same token, these dynamics present a challenge in ensuring that you have the right data, both in terms of quality and quantity.

Building a high-quality filter ensures that the data you end up with is sufficient for your use-case and representative of the conversation you wish to analyze. Moreover, building a proper filter can significantly reduce the time and effort required to clean your dataset and ensure you are working within access limits. Taking time to create a high-quality filter can be one of the most important steps you take to successfully study Twitter data.

In this guide, we walk you through best practices to build filters for your analysis, including:

- Key questions to consider as you decide what data you need
- Important factors to consider that may change or impact your dataset
- New operators available with the new Twitter API
- Examples of using operators to build rules and queries
- Additional factors to consider when collecting data for research and analysis

Table of contents

Scoping for the right data

Types of filtering

Important factors to consider when collecting Tweets

A look at operators and their relevance for research and analysis use-cases

Example queries & rules

Additional Resources

Scoping for the right data

Before you start using any operators or building rules to get Twitter data, it is important to define the scope of your data requirements. This article on 'Do more with Twitter data (https://twitterdev.github.io/do_more_with_twitter_data/finding_the_right_data.html)' highlights some important questions to ask as you plan on what data is required for your use-case. These are explained with additional details below:

Why

This question relates to the purpose of your use-case. Understanding 'why' you are looking for data on a certain topic will help you identify the kind of data you need. For example, the data you may need for a class project on visualization (where a sample of Tweets might be sufficient) will be different from what you might need to train and build a classifier for solving a machine learning (ML) problem (where you might need more, specific data).

What

This question pertains to the kind of data you need. Some points to consider here include:

- Do you need Tweets in a certain language?
- Do you need Tweets with images?
- Do you need Tweets that are retweets?

Answering these questions will help inform your decision on the kind of data you need (and thus the appropriate filtering criteria). Below are some tips to keep in mind as you think of getting data for your use-case:

When

This question helps you scope the time-period of the data you need for your use-case. Determine if the topic you wish to study requires historical data (if yes, then from how far back) or if you need Tweets as they are created in real-time. Based on this you can decide which endpoint will serve your needs.

- For historical data more than 7 days old, you can use the premium Search Tweets: Full-Archive API.
- For data within the last 7 days or recent updates you can use the new recent search endpoint.
- For getting Tweets filtered in real-time, you can use the new filtered stream endpoint.

Who

Based on your use-case, you may be interested in getting Tweets from:

- A specific account
- Followers of a certain account
- An audience or set of accounts (for example, key news organizations)
- Everyone who is talking about a certain topic

This question will help you identify the right source for Tweets for your use-case. The endpoint that you use will also depend on the answer to this question. For example, in order to study certain accounts, you may want to use the user_timeline endpoint, followers/ids etc.

Where

This question pertains to identifying the geographic location of Tweets and Users for your use-case. An example of this is studying the conversation about Hurricane Harvey. For this, researchers and developers of emergency management systems may want to learn how users in Houston were using Twitter during the 2017 hurricane. To understand this, they may look at ‘geo-tagged’ Tweets from Houston or from those users whose location is Houston. More information on ‘geo-filtering’ in the next section.

Types of filtering

When getting data from Twitter APIs, there are a couple of ways of filtering to get the data for your needs.

Filtering using operators

When using the Twitter API, you can use operators to specify the data to filter for. This will limit the data you receive from Twitter APIs to meet the criteria you specify. The more well defined, comprehensive rules and filtering you use, the less likely you are to end up with data irrelevant to your needs (and/or fewer attempts to go back and get data you missed). This will result in reduced efforts cleaning your dataset. You can learn more about the operators available for the recent search endpoint as well as the filtered stream endpoint using the links below:

- Operators for the recent search endpoint
- Operators for the filtered stream endpoint.

Filtering using custom code logic

You might want to filter for Tweets based on advanced conditions that are not supported by the operators directly but require custom coding instead. An example of this might be filtering for Tweets from users that have at least ‘x’ followers. Such criteria requires post-processing. Consume the Twitter data first, and then do the conditional check on the user object.

Important factors to consider when collecting Tweets

Exploratory Analysis

Once you have identified the query /rule for your data needs, instead of starting to collect data storing a large number of Tweets, it makes sense to collect some of this data and do an exploratory analysis on the initial subset of collected Tweets. This will allow you to validate your query/rule to see if you are getting the right data. You can also learn additional things from this exploratory analysis such as:

- What additional terms are being used as part of the conversation that you are studying
- Certain 'slang' terms about the conversation that you may have not included
- Are certain hashtags being used as part of the conversation
- Are people from a certain location Tweeting about the topic more than others
- Are certain media content being used as part of the conversation
- Are people sharing urls etc. as part of the conversation

Based on the initial findings from exploratory analysis, you can then refine your rules and adjust those to ensure the data you get is relevant to your data needs.

Some common ways of doing such exploratory analysis can include:

- If you use Pandas in Python or use R, you can use the head() function to explore the initial few dataframes of Tweets and understand the content better
- Use a visualization library such as matplotlib (Python) or ggplot2 to get a visual representation of the Tweet data to identify patterns in Tweets
- Tokenizing Tweets to count frequency and occurrences of certain keywords

Filtering out noise

As mentioned in an earlier section, if you wanted all Tweets about ironman, the triathlon competition and not ironman the movie, going through initial exploratory analysis would help you identify the fact that the resulting data contains Tweets about ironman, the movie (including different terms of relevance to the movie but not the race). Or you might be looking for Tweets about ‘dogs’ but after the exploratory analysis, you realize that the initial Tweet data includes Tweets about ‘hotdogs’, ‘watch dogs’ (the video game), etc. You can now refine your rule to exclude terms (using the negation operator ‘-’) like ‘hot’ or ‘video game’ to eliminate noise and ensure that the resulting data you get is relevant to your use-case.

Additionally, after the exploratory analysis you may also decide to filter out noise from your data collection based on:

- User Accounts

  You may decide to exclude certain accounts based on their account characteristics or content quality, or you might exclude Tweets from or to a specific account that is irrelevant. You might also do this if certain User accounts are frequently mentioned in conjunction with irrelevant keywords.

- Tweet content

  You may decide to exclude Tweets based on the content itself. For example, if you seek to understand sentiment about a topic, you may want to exclude Tweets that only contain hashtags without any other text, content, or phrases that will help you determine sentiment. In order to refine your data for these Tweets, you will need to write custom logic in your code using regular expressions, etc.

- Retweet behavior

  Maybe you are studying the content of Tweets themselves and the volume/metrics of the amount of Tweets might not be highly relevant for you. In this case, you can then fine tune and filter out Retweets (for example, -is:retweet). Alternatively, if you are interested in the way a piece of information has spread, you may choose to look at Retweets as part of your analysis.

To summarize, use a combination of approaches to filter for noise:

- Use operators to scope the data that you receive directly from Twitter APIs
    - Inclusions, exclusions (negations, ANDs, phrases vs. individual terms)
- Use custom code logic or post-processing as an additional filter based on conditions not met with the operators or other qualitative conditions. Some examples of these include:
    - Tweets from users with certain number of followers
    - Tweets with certain number of Retweets
    - Tweets with certain content structure (for example, exclude all Tweets that are just a hashtag with no additional information)

Use keywords contextually relevant to your use-case

When creating a rule or filtering for data, consider using a variety of terms that are directly relevant to your use-case.

Based on your use-case, you may need to use different terms for the same topic to get Tweets.

- For example, in the US if you use the keyword ‘football’ it will refer to American football and not ‘soccer’ which is referred to as ‘football’ in other parts of the world.
- Similarly, you may be using the term ‘ironman’ to refer to the triathlon competition titled ironman and not the movie or superhero ironman.

So, making sure that you use the right set of contextually relevant terms will help you get the right data.

For most use cases, hashtags alone are not representative of the entire conversation on a topic. It is extremely common for participants in a topical conversation to not include a given hashtag in their Tweets (or they may use or start a different one). For example, you might be interested in the conversation about an on-going topic such as traffic in a certain city, some users may use a hashtag (for example, #nyctraffic) in their Tweets about this topic, but many will not. A hashtag can be a great starting point to help identify all of the other related and/or derivative terms, users and hashtags in a conversation, but it is typically insufficient or not representative of a topic or conversation when used in isolation.

Studying threads and conversations

You may be interested in studying responses and conversations related to a Tweet of interest. Previously, getting threads and conversations including retweets with comments, replies etc. was a bit challenging for developers. However, with the new Twitter API, this process has been simplified with the introduction of the conversation ID feature, which now includes the parent Tweet (for a Tweet which was in reply to it). More information on these fields and how to use them is mentioned later in this article.

Content/Topic Drift

When you start collecting data on a certain topic, you start with an identified set of keywords that might be associated with that topic. But as the conversation evolves, there might be other new keywords that emerge. For example, if you started collecting Tweets about the Coronavirus early in the pandemic, you may use keywords such as covid19, covid-19, corona, coronavirus, etc. But, the conversation evolved to other things related to the Coronavirus which can be identified with new terms such as ‘quarantine’, ‘#stayathome’, ‘social distancing’, etc. As part of your study, you might be interested in getting Tweets about this topic that contain some or all of these new keywords. Below are a couple of strategies on addressing content or topic ‘drift’:

Keeping track of common terms in your data pipeline

The most fundamental approach for identifying shifts in conversation is to keep track of commonly appearing terms and emergence of new terms associated with topics of interest to you. As you build your initial data pipeline to consume and store Twitter data for your use-case, you can maintain metadata for Tweets ingested over a time-period. You can use this metadata to identify emergence of new keywords.

In order to do so, as you ingest data in your pipeline, you can tokenize Tweets to remove stop words, special characters etc. and keep aggregated counts and frequency of words per time period. Using this aggregated data, you can then identify which new terms might be showing up. Once you have identified new terms, you can update your query (if you use recent search) or update your rules (if you use the filtered stream) to include the new terms.

Using Annotations (by using context or entity operators)

The new Twitter API v2 endpoints - filtered stream and recent search - both support filtering by context or entity operators that allow you to get Tweets by topics. Thus, instead of using a variety of different search terms about a topic, you can use the context or entity operators to specify the domain or entity for which you want Tweets about. This way, even if the underlying keywords about a topic changes and if Twitter associates any new terms with that topic, Tweets with those keywords will be included in your response. If you decide that you want to include additional keywords in your dataset, you can always supplement those additional terms in your query or rules. At a minimum, the annotations can serve as a helpful starting point. Learn more about how to use annotations in the section on operators in this article.

Sample & Collection Bias

The way you collect data is important as it can influence your analysis, evaluation, and conclusion. In the context of Twitter data, below are some examples of how bias is introduced, as well as ideas to address bias.

Relying on a narrow set of keywords

If you rely on one or two hashtags or keywords for collecting Tweets on topics that might be happening with various other keywords as well, you may end up with a dataset that might be skewed towards certain conversations. For the coronavirus example mentioned above, if you only used covid19 as a search term, you may end up entirely missing the relevant conversation that uses other words such as quarantine, #stayathome, etc.

Relying on geo-tagged Tweets only

A small percentage of Tweets have geo-tagged information associated with them. Thus, relying only on geo-tagged Tweets alone might introduce some bias in your dataset. You should consider this as part of your data collection methodology and use-case.

Relying on a small dataset

Based on your use-case, you should ensure that your dataset is of sufficient size with Tweets that you think are representative of the conversation. For a ML use-case such as classification, using a very small dataset might lead to overfitting of models, so you should make sure that you include sufficient data for your study. You should not draw conclusions about the entire conversation when using only a small subset of dataset of Tweets.

Relying on Tweets from specific time period only

Whether your use-case requires you to look for data about an event that happened in the past (historical data) or if you are studying a current (on-going) event, carefully consider the time period for which you get the data for in order to ensure that you consider all aspects of the conversation you wish to study. Selecting an incorrect or incomplete time frame (missing key days or the head, tail of the event) may result in bias in your dataset.

Compliance

It is important to keep in mind that if you store Twitter data, you must keep it compliant as part of our developer policy. Learn more on how to do that on our documentation page.

A look at operators and their relevance for research and analysis use-cases

In order to filter Twitter data you need to specify a ‘query’ (if you use recent search endpoint) or create a ‘rule’ (if you use the filtered stream endpoint). These queries and rules are a combination of operators that help you specify the data you need. If you have used our v1.1 search/tweets endpoint before, you will be familiar with the operators for that endpoint. The new endpoints (recent search and filtered stream) supports new operators. Check out our documenatation for a complete list of operators for the new recent search endpoint, and the new filtered stream endpoint.

Note: The way that the fields can be requested for the Tweet object from the recent search endpoint is different from our old search/tweets endpoint. The new Tweet payload also supports additional fields that are not included in the v1.1

search/tweets endpoint. A complete comparison of the old and new Tweet payload can be found on the migration hub.

Below is a list of some of these new operators and their relevance to research and analysis use-cases:

Conversation ID

The recent search endpoint introduced a new operator conversation_id that lets you get all Tweets that are part of a conversation. For example, if you want to retrieve all replies to a Tweet to understand the follow up conversation, you use the conversation_id operator in the recent search query with the Tweet ID of the first Tweet.

Before this operator was available, developers had to use their own techniques to get replies to a Tweet. This included using the 'to' operator and getting all Tweets to a handle and then parsing the responses and pulling out Tweets matching the in_reply_to_status_id_str. The conversation_id operator removes the need to do this custom logic and make it simple to get all Tweets that are part of a conversation (including replies).

Example of using this operator in the section on example queries and rules below.

Tweet annotations (context and entity)

The filtered stream and recent search endpoints both support new 'context' and 'entity' operators that help you get Tweets based on the contextual interpretation of the Tweet data. These operators allow you to get data for topics or entities without writing a lot complex queries with a huge list of search terms. For example, if you want all Tweets about American Football, previously you had to use a variety of terms in your search query or rules such as (NFL OR Football OR "Baltimore Ravens" OR "Seattle Seahawks") etc. With the context operator, you could simply replace the above query with context:11.689566306014617600 and you will get all Tweets about American Football.

The context operator helps filter Tweets by a domain or topic and the entity operator helps filter Tweets by named entities. Both domain and entity are identified by an id and have a name property as well. Examples of domains include Brand, Product, Person etc. Examples of entities include NFL Football, Baltimore Ravens etc. The format of using the context operator is: context:<domain_id>.<entity_id>

You can learn more about Annotations and supported domains on our documentation. Example of using this operator in the section on example queries and rules below.

Precedence of AND before OR

One key difference to keep in mind when using operators with the new recent search and filtered stream endpoints is that

- In the old v1.1 API (search/tweets), OR is applied before logical AND(which is denoted by a space between terms or operators)
- In the new Twitter API (recent search and filtered stream), AND is applied before OR

See example below:

Query: corona covid OR covid-19

Interpretation in old standard search endpoint:

Will return all Tweets with the term corona along with either the term covid or covid-19

Interpretation in new recent search endpoint:

Will return all Tweets that either contain:

- both the terms - corona and covid
- or the term covid-19

Example queries & rules

Here are some examples of using common operators for building rules & filters

Using negation

The negation operator allows you to exclude Tweets based on certain conditions. It is denoted by a hyphen (-) preceding an operator.

For example, If you want to get Tweets about the ironman competition, but not about ironman the movie or superhero, you could exclude those terms using the negation operator as shown below:

ironman -movie -superhero

Similarly, if you want to exclude all Tweets that are not Retweet, you can use:

-is:retweet

Note: Do not negate a set of operators grouped together in a set of parentheses. Instead, negate each individual operator.

For example, Instead of using -(grumpy OR cat OR meme), we suggest that you use -grumpy -cat -meme

Using AND

The AND operator allows for the logical AND between operators. Thus, when AND is used between operators, Tweets that meet all the conditions will be returned.

For example, if you want to get Tweets about both ‘cat’ AND Tweets about ‘dog’, you can format your query as follows:

cat AND dog

In the example above, only Tweets that match on both the term ‘cat’ and the term ‘dog’ will be included

Note: a space between two terms or operators also indicates a logical AND

Using OR

The OR operator allows you to set the logical OR condition between other operators. Thus, if one of the conditions is met for a Tweet, then that Tweet will be returned in your response.

For example, if you want to get Tweets about ‘cat’ or Tweets about ‘dog’, you can format your query as follows:

cat OR dog

In the example above:

- if the Tweet mentions ‘dog’ it will be included in your response.
- if the Tweet mentions cat it will be included in your response.

Exact phrase matching

You can search for Tweets that contain the exact phrases by specifying the phrase within double quotes (“). For example, if you want all Tweets that contain the exact phrase “Twitter API”, a query like:

Twitter API

might not work. This will include Tweets that contain the word Twitter and the word API but not necessarily next to each other as a phrase “Twitter API”. So for example, a Tweet like: “I am new to Twitter and I want to learn about API programming” might be included in your dataset whereas you might be looking for Tweets that talk about the phrase Twitter API like “I am interested in learning about the Twitter API”. In order to get this exact match, you can use the Twitter API in quotes as:

“Twitter API”

Filtering by language

The lang operator allows you to filter for Tweets by language. So, you can specify the language in which you want the Tweets to be in (using the language code). Check out the list of supported languages. The format of this operator is:

lang:language_code

For example, if you want all Tweets in english, your lang operator will be:

lang:en

Using parentheses

You can use parentheses to group operators together. This allows you to organize your query and ensures that the right logical operators are applied to your query. For example:

(grumpy cat) OR (#meme has:images)

will return either Tweets containing the terms grumpy and cat, or Tweets with images containing the hashtag #meme.

Note: AND operator is applied first, before the OR.

Getting replies to a Tweet

You can get replies to a Tweet using the new operator conversation_id. When a Tweet is part of a conversation thread, the conversation ID for that Tweet will match the parent Tweet ID.

An example for using this operator is:

conversation_id:1255542774432063488

Getting Tweets by topic

Instead of using a list of various keywords, you can get Tweets about certain topics when using the recent search or filtered stream endpoints, using two operators:

1. **context**

This operator allows you to get Tweets with a specific domain ID and/or domain ID, enitity ID pair. The list of supported domains can be found on the annotations page. The format of using this operator is as one of the following:

context:domain_id.entity_id [This will give you all Tweets about an entity within that domain]

context:domain_id.* [This will give you all Tweets about that domain]

For example:

To get Tweets about the domain ‘NFL Football Game’, your context operator will be:

context:28.* [28 is the domain id for this domain]

2. **entity**

This operator allows you to get Tweets by using the entity name string. The format of using this operator is as follows:

entity:entity_name [entity name is the string value, name for the entity]

For example:

To get Tweets about the Baltimore Ravens, your entity operator will look like:

entity:Baltimore Ravens [“Baltimore Ravens” is the name of the entity]

Summary of example operators

Operator	Usage
-	Exclude terms or operators that follow the hyphen
AND	Logical AND between terms/operators to include ALL Tweets that match the criteria between the AND
OR	Logical OR between terms/operators to include Tweets that match either one of the criteria between the OR
“”	Use double quotes to specify the exact term that you want to search for
()	Use parentheses to group operators/terms and organize your queries
lang	Specify the language for the Tweets you want
conversation_id	Retrieve the conversation including replies to a Tweet using the Tweet ID
context	Get Tweets for a topic specified using the domain_id or entity_id or both
entity	Get Tweets for a name entity using the entity name

We hope this tutorial is helpful to you in getting started building filters to get Twitter data. Reach out to us on @Twitterdev (https://twitter.com/TwitterDev) or our community forums (https://twittercommunity.com/) with feedback

Additional Resources

- A complete list of operators for the new recent search endpoint
- A complete list of operators for the new filtered stream endpoint
- Check out this tutorial that shows how to listen for important event
- See how others use the Twitter API to understand the public conversation

Ready to build your solution?

Apply for developer access to get started

Apply for access (https://developer.twitter.com/en/apply-for-access)

Was this page helpful?*

😄 😠

Share additional feedback through our **Twitter Developer Platform Feedback Form**!

Submit

Developer policy and terms
Follow @twitterdev (http://twitter.com/twitterdev)
Subscribe to developer news

Cookies (https://help.twitter.com/rules-and-policies/twitter-cookies)
Privacy (https://twitter.com/privacy)
Terms and conditions (https://twitter.com/tos)