Yatcobot Documentation

Release 1.0.0

John Buluba

Jul 12, 2018

Contents:

CHAPTER 1

Getting Started

The best bot for retweeting twitter giveaways!

Do you want **free** stuff?
Do you like to **win**?
Do
you want to participate in twitter **giveaways** but you dont have the time to search and retweet?
Yatcobot is the solution! A bot that search and retweet giveaways automatically!

1.1 Features:

- **Search for new giveaways** *Search and queue tweets to retweet, with a customizable way*
- **Retweet** *Obviously…*
- **Advanced sort** *Prioritize tweets found based on parameters like user defined keywords, age and popularity*
- **Notification** *Can notify you using pushbullet when someone mentions you so you can quickly get your precious gift*

Follow these steps for a quick start

1.1.1 Installation

The easiest way to install is:

```
pip install yatcobot
```

For more installation methods see (See more at *Installation*)

1.1.2 Configuration

Before starting you must get api keys from twitter. You can get these keys from here. (See more at *How to get twitter api keys*)

You must create a config named *config.yaml* and must at least set the api keys. A minimal config.yaml is

```
twitter:
    consumer_key: your_consumer_key
    consumer_secret: your_consumer_secret
    access_token_key: your_access_token
    access_token_secret: your_access_token_secret
```

You can edit config.sample.yaml that is placed in the root of the project (*dont forget to rename it to config.yaml*) or view the sample on github

Config file is loaded automatically from specific paths. The paths that are searched for config.yaml are (from highest priority to lowest):

1. ***./config.yaml*** Search for config in the current working directory
2. ***~/.config/Yatcobot/config.yaml*** Search in config folder. If for example your username is *user* the full path will be */home/user/.config/Yatcobot/config.yaml*
3. ***default*** The default config that is packaged with the bot.

Also you can define another config with the **–config** argument, which will have the highest priority

Higher priority configs override settings that are defined in the lower. So in your config you only need to define the changes. (See more at *Config options*)

1.1.3 Run

In the directory where your config.yaml run:

```
yatcobot
```

Or you can specify the path of config with:

```
yatcobot --config=/path/to/config.yaml
```

For more command line options (See more at *Command Line Arguments*)

Now just wait for all the free gifts!

1.1.4 Documentation

For more info visit our docs: https://readthedocs.org/projects/yatcobot/.

1.1.5 Donate

Please donate to support the development if you find this software useful (*or if you are so rich from the staff you won*).

CHAPTER 2

Installation

2.1 Source

To use from source:

```
sudo apt-get install git python3 python3-pip
git clone https://github.com/buluba89/Yatcobot.git
sudo pip3 install -r requirements.txt
```

Run using:

```
cd /path/to/repo/
python3 yatcobot.py
```

Or if you dont want to clutter you global packages use **virualenv**

```
sudo apt-get install git python3 python3-pip python-virtualenv
git clone https://github.com/buluba89/Yatcobot.git
cd Yatcobot
virtualenv -p /usr/bin/python3 env
source env/bin/activate
pip3 install -r requirements
```

And run with

```
cd /path/to/repo/
source env/bin/activate
python3 yatcobot.py
```

For more command line options (See more at *Command Line Arguments*)

2.2 Pip

Yatcobot is distributed as a pip package. To install with pip:

```
pip3 install yatcobot
```

Pip installs the package in your PATH you can call *yatcobot* from anywhere

2.3 Usage with Docker

To run container use like below

```
$ docker run -v /path/to/config.yaml:/yatcobot/config.yaml buluba89/yatcobot
```

where /path/to/config.yaml is the path of your config.json

CHAPTER 3

Config options

The config file written in yaml. If some values are not in the config *reasonable* hardcoded defaults are used

Example config with default values:

```
twitter:

  # Client settings
  # consumer key for twitter api
  consumer_key: null
  # consumer secret for twitter api
  consumer_secret: null
  # access token key for twitter api
  access_token_key: null
  # access token secret for twitter api
  access_token_secret: null

  # The bot will stop when api calls remaining are under
  # min_ratelimit_percent of the max that twitter allows
  min_ratelimit_percent: 10


  # Search for contest settings
  search:

    # Queries to use for searching giveaways.
    # You can set the language for a specific query like this
    # - query
    # - query with language filter:
    #   lang: en
    queries:
      - RT to win
      - Retweet and win
      - Giveaway retweet:
        lang: en
      - "using #hashtags need quotes"
```

(continues on next page)

(continued from previous page)

```
    # Will skip all retweeted post until it finds a new one
    # Be ware, that it may use many api calls
    skip_retweeted: false

    # Max tweets that holds the bot in memory to post
    max_queue: 100

    # The maximum quotes that will be recursively search
    # for the original tweet
    max_quote_depth: 20
    # Some tweets are quotes of another tweet
    # This is the mimimum similary between the quote and the post
    min_quote_similarity: 0.5

    #Plugins to filter tweets
    filter:
      # Filter out tweets that have less retweets than `number`
      min_retweets:
        # If this filter method is enabled
        enabled: true
        # If a tweet has less retweets than this number, dont retweet
        number: 20
      # Filter out tweets that contains some criteria
      blacklist:
        # If this filter method is enabled
        enabled: false
        # keywords that if a posts contains would be filtered
        # leave it empty if you want to disable keywords
        keywords: ["obama", "trump"]
        # Filter based on user screen name. Add here the screen names that you want to filter
        # the screen name is the @User name without the @
        users: ["BotSpotterBot"]
    #Plugins to sort tweets
    sort:
      # Give priority to posts that contain some words
      by_keywords:
        # If this sorting method is enabled
        enabled: true
        # Keywords to search
        keywords: ["ps4", "pc"]
      # Give priority to newer posts
      by_age:
        # If this sorting method is enabled
        enabled: true
      # Give priority to posts with more retweets
      by_retweets_count:
        # If this sorting method is enabled
        enabled: true


  # Actions that some giveaways require to enter
  actions:
    # Follow the user that posts the giveaway
    follow:
      # If this action is enabled
```

(continues on next page)

(continued from previous page)

```
      enabled: true
      # Keywords to search in post for follow action
      keywords: ["follow", "follower"]
      # When max_following is reached, will unfollow oldest follows
      max_following: 1950
      # Follow multiple accounts from a single post
      # Experimental feature, please open issue with example post id if unusual␣
↪behavior is observed
      multiple: false
    # Favorite the post
    favorite:
      # If this action is enabled
      enabled: true
      # Keywords to search in post for favorite action
      keywords: ["fav", "favorite", "like"]
    #This action allows to tag friends when requested
    tag_friend:
      # If this action is enabled
      enabled: false
      # Friends usernames to tag. Bot will randomly pick the required number
      # Usually maximum number of required tags is 3, so better define here 3 or more
      friends: ["friend1", "friend2", "friend3"]
      # keywords of tag to serach in post
      tag_keywords: ["tag"]
      # keywords of friend to search in post
      friend_keywords: ["friend", "friends"]
      # keywords of numbers to search in post
      number_keywords:
        1: ["one", "1", "a", "your"]
        2: ["two", "2"]
        3: ["three", "3"]
        4: ["four", "4"]

  # Intervals of bot tasks
  scheduler:
    # How often will search for new posts
    search_interval: 5400
    # How often will retweet
    retweet_interval: 600
    # A random margin from retweet interval to avoid bot detection
    retweet_random_margin: 60
    # Update blocked users list so posts of them are not retweeted
    blocked_users_update_interval: 300
    # How often will delete oldest posts in queue
    clear_queue_interval: 60
    # How often will update the remaining api rate limits
    rate_limit_update_interval: 60
    # How often will check for new mentions
    check_mentions_interval: 600

# Notifiers will notify when somenone mentions the user.(Possible win)
notifiers:
  # Pushbullet notifier
  pushbullet:
    # If the notifier is enabled
    enabled: false
    # Pushbullet api token
```

(continues on next page)

(continued from previous page)

```
    token: my_pushbullet_token
  # Email notifier
  mail:
    # If the notifier is enabled
    enabled: false
    # Email provider smpt server
    host: smtp.provider.com
    # Port
    port: 25
    # Use tls
    tls: false
    # Email username
    username: sender_address@provider.com
    # Email password
    password: my_secure_password
    # Notification recipient
    recipient: sender_address@provider.com
```

Config file is loaded automatically from specific paths. The paths that are searched for config.yaml are (from highest priority to lowest):

1. ***./config.yaml*** Search for config in the current working directory
2. ***~/.config/Yatcobot/config.yaml*** Search in config folder. If for example your username is *user* the full path will be */home/user/.config/Yatcobot/config.yaml*
3. ***default*** The default config that is packaged with the bot.

Also you can define another config with the **–config** argument, which will have the highest priority

Higher priority configs override settings that are defined in the lower. So in your config you only need to define the changes.

3.1 Global

3.1.1 consumer_key, consumer_secret, access_token_key, access_token_secret

The twitter api keys that are needed for interacting with the twitter api. Obtain them from here

3.2 Search

Here are defined all the search related settings

3.2.1 queries

These are the queries that are used to find contests from the twitter. It works like the twitter search bar, so you can experiment there first

Queries are defined as a sequence. It can be strings or mapppings with additional option

For example

```
search:
    queries:
      - RT to win
      - retweet giveaway
      # You can set a language option for a query
      - Διαγωνισμ:
        lang: el
```

3.2.2 skip_retweeted

If enabled the bot will try to find a non retweeted post in the queue until it finds one and retweet it. Beware that for every check an api call is required, so it may be a burst of api calls reaching the ratelimit.

3.2.3 max_queue

The maximum number of tweets that are in the queue to be retweeted. If queue is bigger, some will be deleted. (*seconds*)

3.2.4 max_quote_depth

Some posts are quotes that quoting other quote(..that quoting other quote). So we need to follow the quotes to find the original post that is the contest. This value defines the max quotes that we will follow to get the original post

3.2.5 min_quote_similarity

When the bot gets new tweets, it checks if they are a quote of a contest (some people quote the contest, they dont retweet them). To get rid of that, the similarity between the quote and the post is compared. This is the threshold which we get the quoted tweet as the contest and not the one we got. 1.0 means 100% the same

3.2.6 filter

Plugins to filter out some tweets are defined here

min_retweets

With this plugin we can filter out tweets below a minimum number of retweets

enabled if this plugin is enabled

number Below this number tweets will be filtered out

blacklist

Blacklist posts based on keywords

enabled if this plugin is enabled

keywords If any of these keywords is found in the post, it will be filtered

users Posts from these users will be filtered. Add here the screen names of users, for example for user @User you insert User (without the @)

3.2.7 sort

Plugins to sort the posting queue, so we can prioritize tweets that are more interesting

by_keywords

Give priority to tweets that contain some keywords

enabled if this plugin is enabled

keywords These keywords are used to promote contests that contain this keywords so the bot enters more contests that the user is interested in

by_age

Most recent tweet will get get priority over old ones

enabled if this plugin is enabled

by_retweets_count

Tweets with more retweets will get priority

enabled if this plugin is enabled

3.3 Actions

Here are defined all the action settings. Actions are requests that contests have, like follow and fovorite

3.3.1 follow

enabled

If the follow action is enabled

keywords

These keywords are searched inside the tweet's text to determinate if it is needed to follow the original poster.

max_following

After this number of following users is reached, will start to unfollow the oldest follows

multiple

When this option is enabled, all users that are mentioned in the post will be followed

Warning: *this is an experimental feature. Please open issue with example post id if unusual behavior is observed*

3.3.2 favorite

enabled

If the favorite action is enabled

keywords

These keywords are searched inside the tweet's text to determinate if it is needed to favorite the original post.

3.3.3 tag_friend

enabled

If the tag friend action is enabled

friends

The usernames of friends that will be tagged. A random username from this list is selected every time, so the more you add the better. **At least 3 are recomended** (giveaways dont ask more than 3 tags usually)

tag_keywords

Keywords for searching 'tag' in post. Useful for international posts

friend_keywords:

keywords for searching 'friend' in post. Useful for international posts

number_keywords

keywords for mapping words to number for identifying how many friends to tag. Useful for international posts

3.4 Scheduler

Intervals of bot tasks

3.4.1 search_interval

How often will search for new tweets from twitter. (*seconds*)

3.4.2 retweet_interval

How often a retweet will be posted. (*seconds*)

3.4.3 retweet_random_margin

Adds randomness to the post interval. For example if retweet_interval is 600 and retweet_random_margin is 60, retweets will be posted every 9-11 minutes. (*seconds*)

3.4.4 blocked_users_update_interval

The interval to update the twitter blocked users so you dont retweet posts from unwanted users. (*seconds*)

3.4.5 clear_queue_interval

How often the queue will be checked so if the number is over max_queue, delete some posts. (*seconds*)

3.4.6 rate_limit_update_interval

How often will update for the remaining api calls

3.4.7 check_mentions_interval

How often we check if the user is mentioned in a tweet. We check this because many contests mention the winners in a tweet, so we can notify the user for a possible win.

3.5 Notifiers

Notifiers are used to notify user when someone mentions him. This usualy implies that the user won something.

3.5.1 pushbullet

Use push bullet for sending notifications to pc or phone

enabled

if this notifier is enabled

token

the pushbullet api token

mail Mail notifications !

enabled

if this notifier is enabled

host

the hostname of the smtp server

port

port of the smtp server

tls

enable tsl encryption

username

username to login to the smtp server

password

password for login

recipient

email address of the notification recipient

CHAPTER 4

Command Line Arguments

Yatcobot supports a number of command line arguments for configuration during launch

```
usage: yatcobot.py [-h] [--login] [--config CONFIG]
                   [--ignore_list IGNORE_LIST] [--log LOGFILE] [--debug]

Yatcobot: a bot for entering twitter contests

optional arguments:
  -h, --help            show this help message and exit
  --config CONFIG, -c CONFIG
                        Path of the config file
  --ignore_list IGNORE_LIST, -i IGNORE_LIST
                        Path of the ignore file
  --log LOGFILE         Path of log file
  --debug               Enable debug
  --test-mail           Test mail settings
```

-h, --help Prints help with all the available command line options

--config, -c **config.yaml by default**. The path of the configuration file that will be used.

Example usage:

```
yatcobot --config /path/to/config.yaml
```

--ignore_list, -i **ignorelist by default**. The path of the ignore_list file, that stores the tweets that have been retweeted.

Example usage:

```
yatcobot.py --ignore_list /path/to/ignorelist
```

--log, -l Enables logging output to a file. Example usage:

```
./yatcobot --log /path/to/out.log
```

--debug	Enables verbose output. Used for debug purposes
--test-mail	Send a test email to your mail notfication recipient

CHAPTER 5

How to get twitter api keys

Comming soon

CHAPTER 6

Donate

Please donate to support the development if you find this software useful (*or if you are so rich from the staff you won*)

Flattr

CHAPTER 7

Licence

This program is released under GPL v2

CHAPTER 8

Getting Started

The best bot for retweeting twitter giveaways!

Do you want **free** stuff?
Do you like to **win**?
Do
you want to participate in twitter **giveaways** but you dont have the time to search and retweet?
Yatcobot is the solution! A bot that search and retweet giveaways automatically!

8.1 Features:

- **Search for new giveaways** *Search and queue tweets to retweet, with a customizable way*
- **Retweet** *Obviously. . .*
- **Advanced sort** *Prioritize tweets found based on parameters like user defined keywords, age and popularity*
- **Notification** *Can notify you using pushbullet when someone mentions you so you can quickly get your precious gift*

Follow these steps for a quick start

8.1.1 Installation

The easiest way to install is:

```
pip install yatcobot
```

For more installation methods see (See more at *Installation*)

8.1.2 Configuration

Before starting you must get api keys from twitter. You can get these keys from here. (See more at *How to get twitter api keys*)

You must create a config named *config.yaml* and must at least set the api keys. A minimal config.yaml is

```
twitter:
    consumer_key: your_consumer_key
    consumer_secret: your_consumer_secret
    access_token_key: your_access_token
    access_token_secret: your_access_token_secret
```

You can edit config.sample.yaml that is placed in the root of the project (*dont forget to rename it to config.yaml*) or view the sample on github

Config file is loaded automatically from specific paths. The paths that are searched for config.yaml are (from highest priority to lowest):

1. ***./config.yaml*** Search for config in the current working directory
2. ***~/.config/Yatcobot/config.yaml*** Search in config folder. If for example your username is *user* the full path will be */home/user/.config/Yatcobot/config.yaml*
3. ***default*** The default config that is packaged with the bot.

Also you can define another config with the **–config** argument, which will have the highest priority

Higher priority configs override settings that are defined in the lower. So in your config you only need to define the changes. (See more at *Config options*)

8.1.3 Run

In the directory where your config.yaml run:

```
yatcobot
```

Or you can specify the path of config with:

```
yatcobot --config=/path/to/config.yaml
```

For more command line options (See more at *Command Line Arguments*)

Now just wait for all the free gifts!

8.1.4 Documentation

For more info visit our docs: https://readthedocs.org/projects/yatcobot/.

8.1.5 Donate

Please donate to support the development if you find this software useful (*or if you are so rich from the staff you won*).

Flattr