Twatbot Documentation

Release 0.0.1

Simo Linkola

September 18, 2014

CONTENTS

Contents:

CHAPTER

ONE

TWATBOT

TwatBot is a twitter bot built for a course "Computational foundations of linguistic creativity" held in University of Helsinki, fall 2014.

TwatBot generates color names based on the knowledge it has on different colors, and their relations.

1.1 Weekly reports

Here are some comments on weekly development during the course.

1.1.1 Week 2

I have made some color utilities into the `colorbot/color_utilities.py`, otherwise the project is still a clean Django-project skeleton. The documentation of the project can be found here. Most of my weeks work has went to setting up and learning Sphinx (and how to hook it to be nearly automatically served in github pages) as I haven't used it before and now seemed like a good time to start using it!

CHAPTER

TWO

MODELS

Django models for resources given on the course.

class color_tweets.models.**BracketedColorBigrams**(**args*, ***kwargs*)

Bracketed bigrams.

Fields:

start_bracket (CharField): starting bracket, max_length = 40.

w1 (CharField): first word, max_length = 40.

w2 (CharField): second word, max_length = 40.

end_bracket (CharField): ending bracket, max_length = 40.

f (PositiveIntegerField): bigram's frequency

Sample entries:

start_bracket	w1	w2	end_bracket	f
and	cheddar	cheese	,	3477
the	eye	candy	.	3476
with	wood	wool	and	3448

class color_tweets.models.**ColorMap**(**args*, ***kwargs*)

Color stereotype - rgb (in html format) value pairs with base color names.

Fields:

stereotype (CharField): Name of the color stereotype, max_length = 40.

color (CharField): Base color, max_length = 40.

html (CharField): Color in html-format, i.e. #rrggbb, where r, g and b are hex codes.

Sample entries:

stereotype	color	html
acid	green	#B0BF1A
absinthe	green	#7FDD4C
acorn	brown	#7F6241

class color_tweets.models.**ColorUnigrams**(**args*, ***kwargs*)

Color unigrams.

Fields:

solid_compound (CharField): Solid compound of the unigram, max_length = 50.

f (PositiveIntegerField): unigram's frequency

Sample entries:

solid_compound	f
aluminumleather	594
amberbunny	240
amberdawn	300

class color_tweets.models.**EveryColorBotTweets**(**args*, ***kwargs*)
URL's and hex color codes for Everycolorbot's tweets.

Fields:

hex (CharField): Color in hex-format, i.e. 0xrrggbb, where r, g and b are hex codes.
url (URLField): URL for the tweet.

Sample entries:

hex	url
0x634ef9	http://t.co/9OXdTPFOXK
0x31a77c	http://t.co/99kdUpov9E
0x98d3be	http://t.co/Os53Hh3qs7

class color_tweets.models.**PluralColorBigrams**(**args*, ***kwargs*)
Plural color bigrams.

Fields:

w1 (CharField): first word, max_length = 40.
w2 (CharField): second word, max_length = 40.
singular (CharField): singular of the second word, max_length = 40.
f (PositiveIntegerField): bigram's frequency

Sample entries:

w1	w2	singular	f
bile	salts	30370	salt
tree	leaves	30015	leaf
maple	leafs	29701	leaf

class color_tweets.models.**UnbracketedColorBigrams**(**args*, ***kwargs*)
Unbracketed bigrams.

Fields:

w1 (CharField): first word, max_length = 40.
w2 (CharField): second word, max_length = 40.
f (PositiveIntegerField): bigram's frequency

Sample entries:

w1	w2	f
summer	storm	2302
lobster	bisque	2284
tan	leather	2282

CHAPTER

THREE

COLOR UTILITIES

Utility functions for working with colors.

Supports three types of color definitions:

- hex: `str` in form of `0xrrggbb`, where `r`, `g` and `b` are hex codes.
- html: `str` in form of `#rrggbb`, where `r`, `g`, `b` are hex codes.
- rgb: `tuple` in form of `(r, g, b)`, where `r`, `g` and `b` are integers in [0, 255].

Note: Functions defined in this module are ignorant of semantics, and you should do your reasoning, e.g. what to blend and how to blend, before calling functions in this module.

`color_tweets.color_utils.`**`blend`**(*head*, *modifier*, ***kwargs*)

Blend two colors.

Blending of colors is done based on optional keyword arguments that are given to the function. If no arguments are given, blending is a mean of the rgb-values of input colors.

Args:

head: head color in any supported format.
modifier: modifier color in any supported format.
kwargs: Optional blending instructions. Currently supported keyword arguments are:

a_head (`float`): amount of head color to mix. Should be in [0, 1].
a_rgb (`tuple`): amount of each head color component to
mix, each value in tuple should be in [0, 1].

If a_rgb is present, a_head is ignored.

Returns: Blended color as rgb-tuple.

`color_tweets.color_utils.`**`dist`**(*color1*, *color2*)

Calculate distance between two colors.

Currently the distance is calculated between euclidean distance between rgb-tuples. Input colors can be in any supported format.

Args:

color1: first color
color2: second color

Returns: Distance between colors as `float`.

`color_tweets.color_utils.`**`hex2rgb`**(*hex*)
Convert hex-string color into rgb-tuple.

Args: hex (`str`): Color in hex-format, e.g. `0xffeedd`.

Returns: Color in rgb as 3-tuple, e.g. `(255, 255, 255)`.

`color_tweets.color_utils.`**`html2rgb`**(*html*)
Convert html color format string into rgb-tuple.

Args: html (`str`): color in html-format, e.g. `#ffeedd`

Returns: Color in rgb as 3-tuple, e.g. `(255, 255, 255)`.

`color_tweets.color_utils.`**`is_hex`**(*hex*)
Verify that color variable is in accepted hex-format.

Accepted hex-format is a string in form of `0xrrggbb`, where `r`, `g` and `b` are hex codes.

Args: hex (`str`): Variable to be verified.

Returns: `True` if variable is in hex-format, `False` otherwise.

`color_tweets.color_utils.`**`is_html`**(*html*)
Verify that color variable is in accepted html-format.

Accepted html-format is a string in form of *#rrggbb‘*, where `r`, `g` and `b` are hex codes..

Args: html (`str`): Variable to be verified.

Returns: `True` if variable is in html-format, `False` otherwise.

`color_tweets.color_utils.`**`is_rgb`**(*rgb*)
Verify that color variable is in accepted rgb-format.

Accepted rgb-format is an integer 3-tuple with all values in [0, 255]

Args: rgb (`tuple`): Variable to be verified.

Returns: `True` if variable is in rgb-format, `False` otherwise.

`color_tweets.color_utils.`**`rgb2hex`**(*rgb*)
Convert rgb-tuple into hex-color string.

Args: rgb (`tuple`): Color in rgb-format, e.g. `(255, 255, 255)`.

Returns: Color as hex-string, e.g. `0xffeedd`.

`color_tweets.color_utils.`**`rgb2html`**(*rgb*)
Convert rgb-tuple into html color format string.

Args: rgb (`tuple`): color in rgb-format, e.g. `(255, 255, 255)`

Returns: Color in html-format string, e.g. `#ffeedd`.

CHAPTER

FOUR

COLOR TWEETS

Main package of the project, developed as a Django app.

Package has currently following modules:

- **Basic Django app modules**
  - `color_tweets.models` - Django models for resources given on the course.
  - `color_tweets.views` - Custom Django views for the app.
- **Modules special for Color Tweets**
  - `color_tweets.color_utils` - Utility functions for working with colors.

CHAPTER

FIVE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

c

m

B

C

D

E

H

I

M

P

R

U