



Gazouilloire : Collecter des données dans la mare de tweets

Benjamin Ooghe-Tabanou

► To cite this version:

Benjamin Ooghe-Tabanou. Gazouilloire : Collecter des données dans la mare de tweets. Séminaire Morning Dev' (MorDev), Laboratoire de linguistique formelle, Université Paris Cité, Apr 2019, Paris, France. pp.20. hal-03904041

HAL Id: hal-03904041

<https://sciencespo.hal.science/hal-03904041>

Submitted on 16 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License



Equipex DIME-SHS
ANR-10-EQPX-19-01

Gazouilloire

Collecter des données dans la mare de tweets

Laboratoire de Linguistique Formelle – MorDev

16 avril 2019

Benjamin Ooghe-Tabanou

Sciences Po médialab – DIME Web

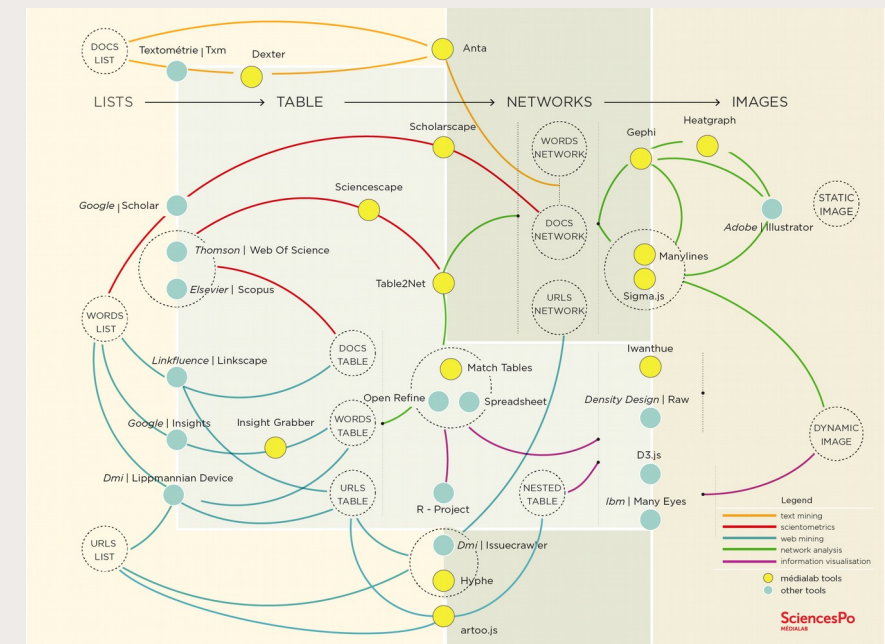
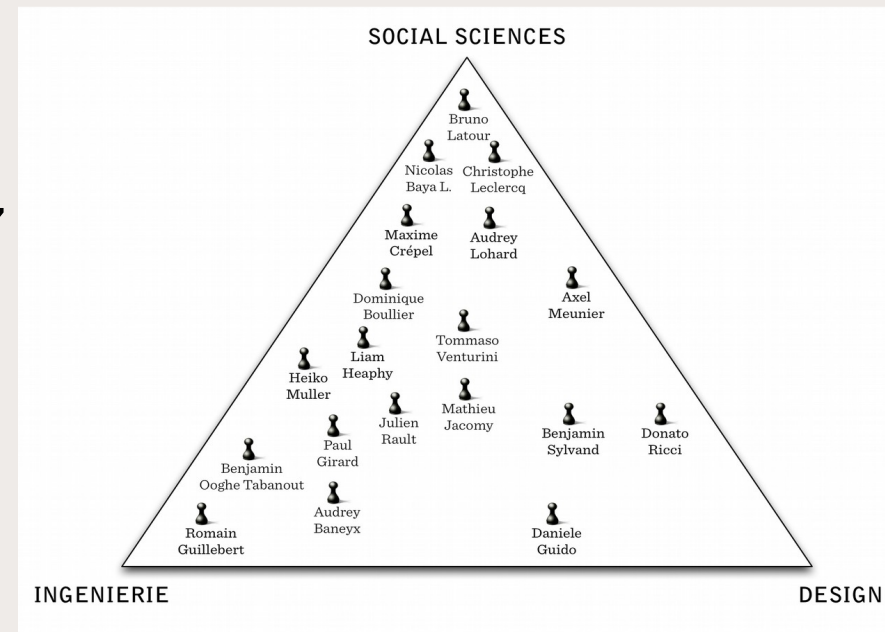
SciencesPo
MÉDIALAB

SciencesPo
DIME-SHS

Le médialab de Sciences Po

<https://medialab.sciencespo.fr>

- Centre de recherche SHS à Sciences Po, fondé par Bruno Latour en mai 2009, dirigé par Dominique Cardon depuis 2017
- Numérique, sciences sociales et design
→ **Interdisciplinarité**
- Articulation des méthodes **quali & quanti**
- Étude des **traces numériques**
- Un écosystème **d'outils**
<http://tools.medialab.sciences-po.fr>
- Un atelier ouvert mensuel : le METAT
<http://www.medialab.sciences-po.fr/atelier/>

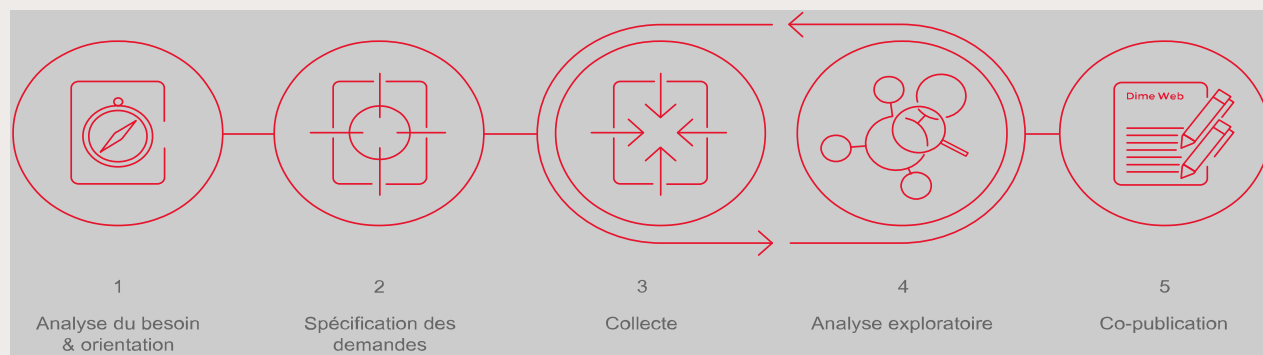
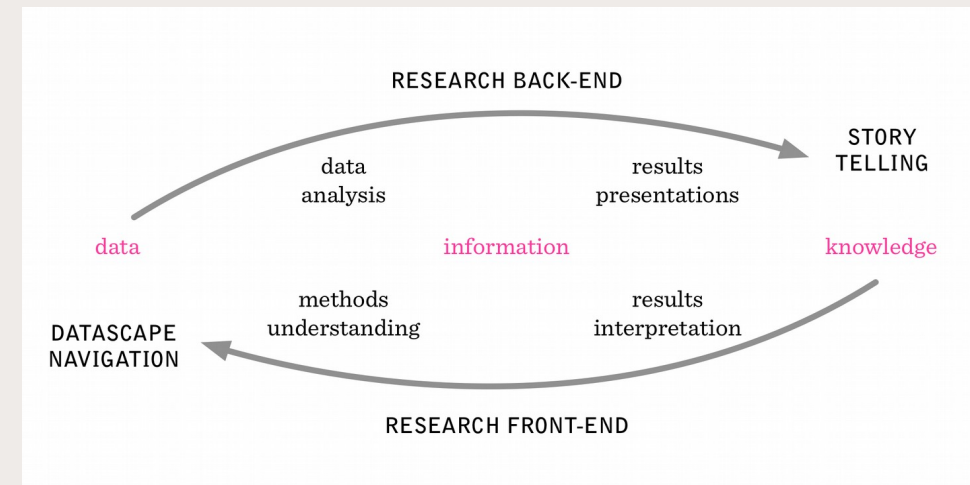


L'instrument DIME Web

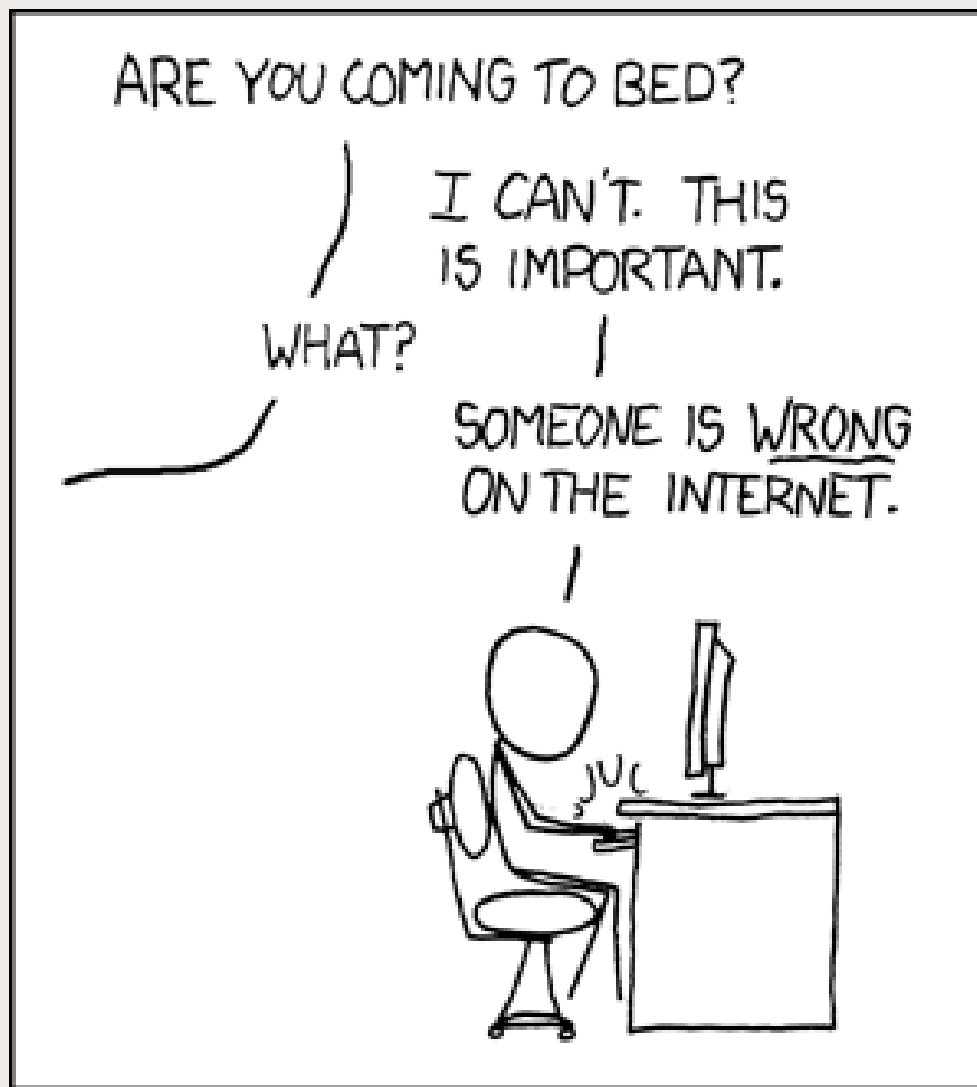
<http://dimeweb.dime-shs.sciences-po.fr>

Accompagnement technique et méthodologique à l'utilisation du Web comme nouveau terrain d'enquêtes

- Collecter, enrichir, nettoyer, visualiser et analyser des traces numériques
- Analyse de réseaux, archivage du web, analyse de controverses (ANT)
- Développement d'outils génériques libres et open source
- Extraction ciblée de contenus
- Analyse Exploratoire de Données
- Méthodes numériques & itératives
≠ tout automatique



Twitter : un espace de dialogue et débat



CC-BY-NC - Randall Munroe - XKCD

Accès contrôlé via les APIs officielles

- **Authentification nécessaire**

<https://developer.twitter.com/en/apps>

- **API gratuite :**

- recherche du passé jusqu'à 7 jours
- nombre d'appels limités

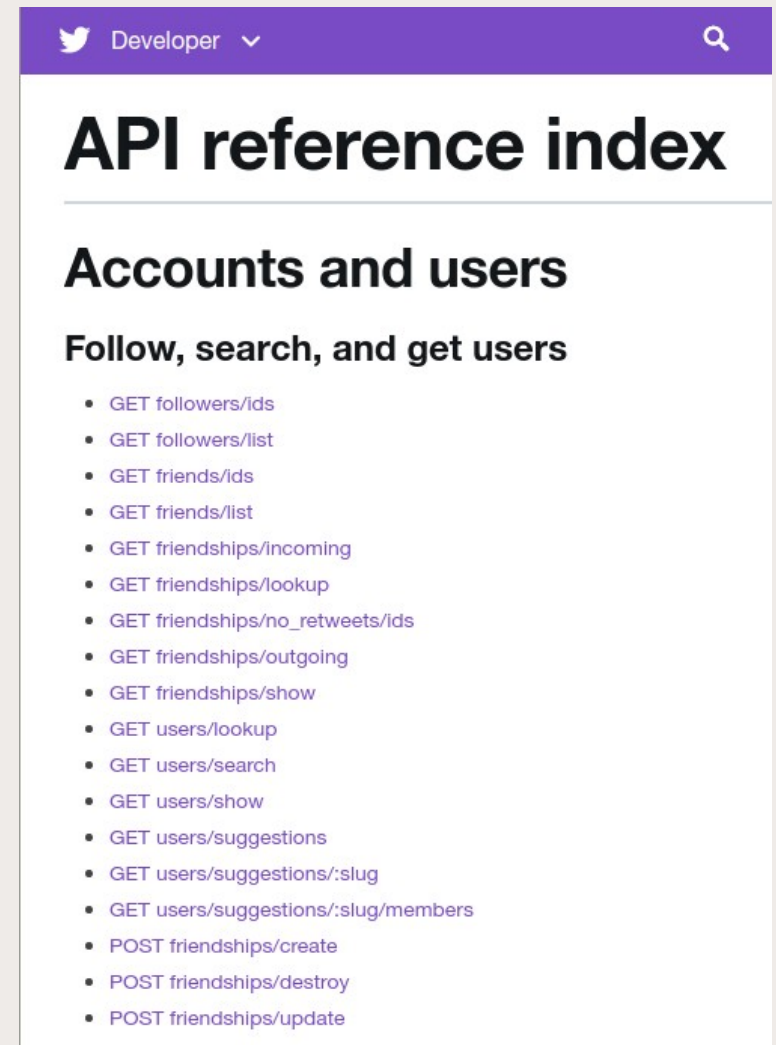
- **API payante :**

- plus riche
- plus de fonctionnalités
- très chère !

- **Différents types d'accès :**

- par ID, user, mot-clé, etc. via **REST API**
- suivi live via **Streaming API**

<https://developer.twitter.com/en/docs/api-reference-index>



Appeler l'API en programmant en Python

<https://github.com/sixohsix/twitter>

```
from twitter import Twitter, OAuth

t = Twitter(auth=OAuth(token, token_secret, consumer_key, consumer_secret))

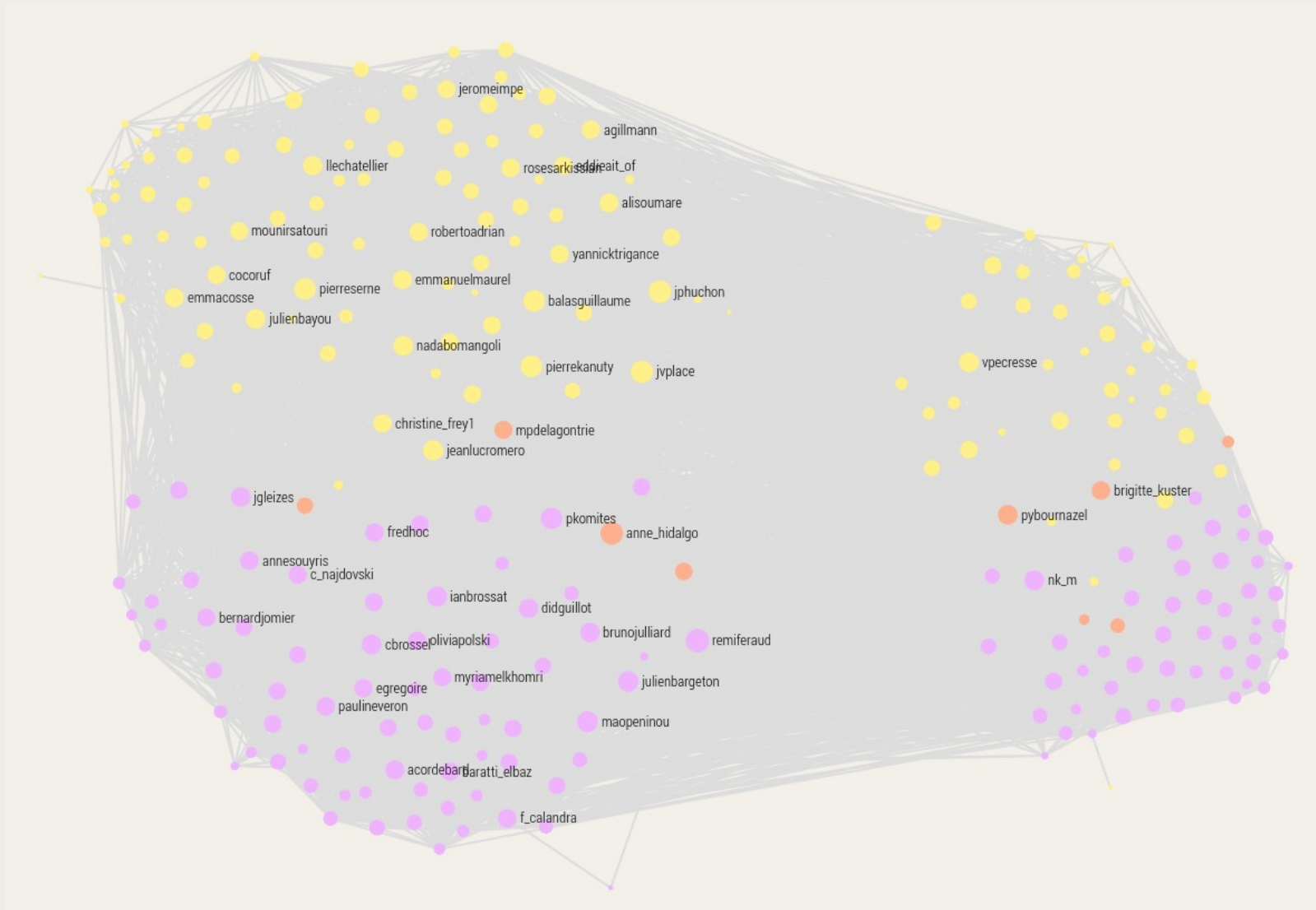
t.statuses.home_timeline(count=10)
t.statuses.user_timeline(screen_name="medialab_ScPo")

t.statuses.update(status="I'm sending tweets programatically in Python!")

for tweet in t.search.tweets(q="linguistics", tweet_mode='extended'):
    print(tweet['user']['screen_name'], tweet['text'])
```

- Helpers dans gazouilloire pour différents types d'analyses :
 - collecte des tweets d'un jeu d'users
 - collecte de métadonnées sur un corpus d'users
 - collecte de retweeters d'un ensemble d'users
 - analyse de communautés (followers, mentions, retweets, etc.)
- Outrepasser les limites de l'API via le scraping
<https://github.com/Jefferson-Henrique/GetOldTweets-python>

Explorer des réseaux de followers



Liens de proximité Twitter entre les élus du Conseil Régional d'Île-de-France
et du Conseil de Paris

Gazouilloire : extraction systématique continue

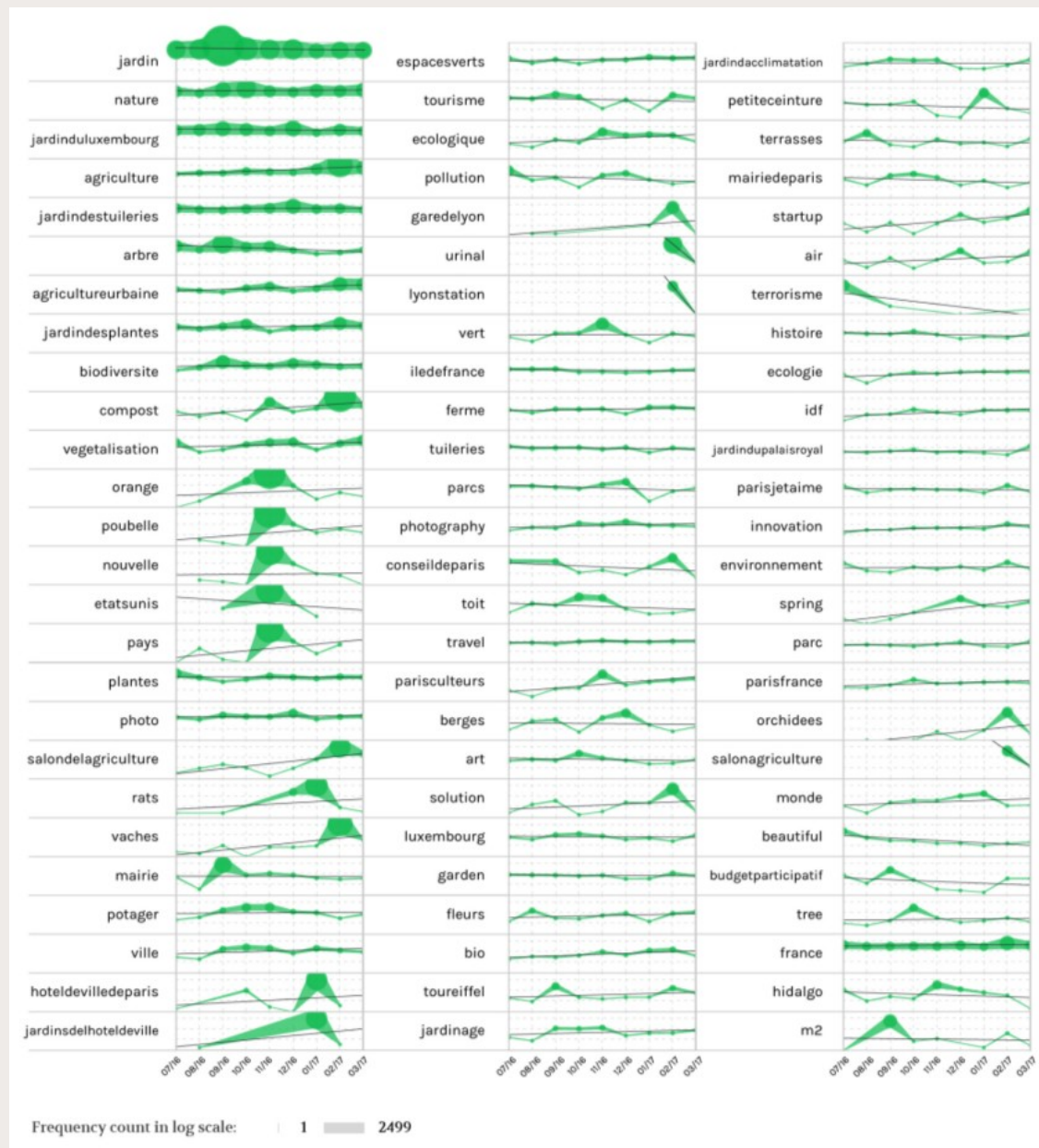
<https://github.com/medialab/gazouilloire>

- Collecter en direct et en continu (et jusque 7 jours en arrière)
 - les tweets citant des mots-clés / hashtags
 - les tweets contenant certains morceaux d'urls
 - les tweets de certains utilisateurs ainsi que ceux les mentionnant
- Filtrer par langue, période temporelle et/ou géolocalisation
- Remonter le fil des conversations
- Collecter les médias embarqués dans les tweets (images et vidéos)
- Résoudre les redirections des urls partagées
- Exporter des fichiers tableurs de métadonnées, de textes
- Calculer des agrégats (sites ou urls partagés, etc.)

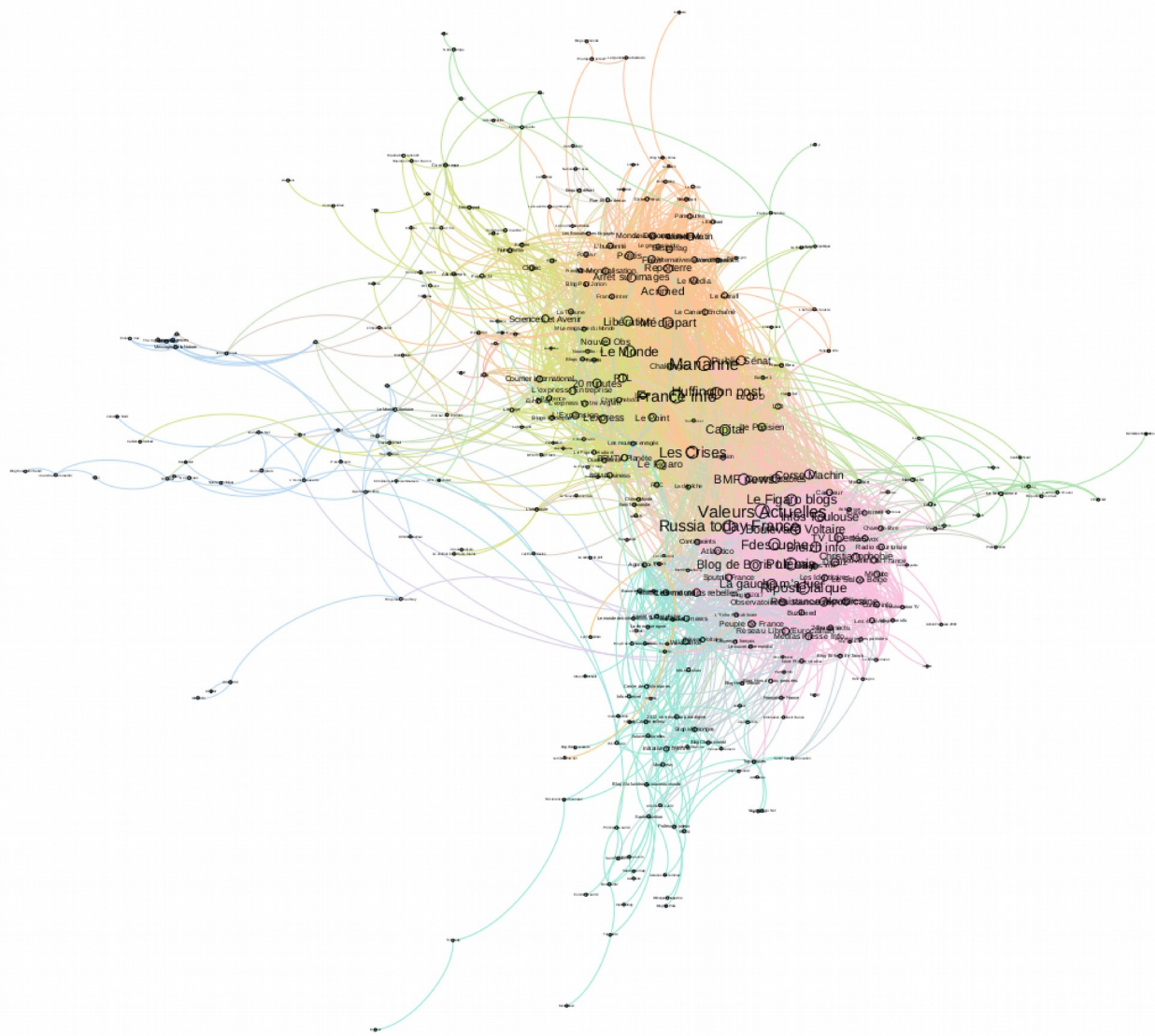
De nombreuses métadonnées à exploiter

```
TWEET_FIELDS = [
    "id",                # digital ID
    "time",              # UNIX timestamp of creation
    "created_at",        # ISO datetime of creation
    "from_user_name",     # author's user text ID (@user)
    "text",              # message's text content
    "filter_level",      # internal TCAT field, ignorable
    "possibly_sensitive", # whether a link present in the message might contain sensitive content according to Twitter
    "withheld_copyright", # whether the tweet might be censored by Twitter following copyright requests, ignorable
    "withheld_scope",    # whether the content withheld is the "status" or a "user", ignorable
    "withheld_countries", # list of ISO country codes in which the message is withheld, separated by |, ignorable
    "truncated",         # whether the tweet is bigger than 140 characters, obsolete
    "retweet_count",     # number of retweets of the message (at collection time)
    "favorite_count",    # number of likes of the message (at collection time)
    "reply_count",       # number of answers to the message, dropped by Twitter (since Oct 17, now charged), unreliable and ignorable
    "lang",              # language of the message automatically identified by Twitter's algorithms (equals "und" when no language could be detected)
    "to_user_name",      # text ID of the user the message is answering to
    "to_user_id",        # digital ID of the user the message is answering to
    "in_reply_to_status_id", # digital ID of the tweet the message is answering to
    "source",            # medium used by the user to post the message
    "source_name",       # name of the medium used to post the message
    "source_url",        # link to the medium used to post the message
    "location",          # location declared in the user's profile (at collection time)
    "lat",               # latitude of messages geolocalized
    "lng",               # longitude of messages geolocalized
    "from_user_id",      # author's user digital ID
    "from_user_realname", # author's detailed textual name (at collection time)
    "from_user_verified", # whether the author's account is certified
    "from_user_description", # description given in the author's profile (at collection time)
    "from_user_url",     # link to a website given in the author's profile (at collection time)
    "from_user_profile_image_url", # link to the image avatar of the author's profile (at collection time)
    "from_user_utcoffset", # time offset due to the user's timezone, dropped by Twitter (since May 18), ignorable
    "from_user_timezone", # timezone declared in the user's profile, dropped by Twitter (since May 18), ignorable
    "from_user_lang",    # language declared in the user's profile (at collection time)
    "from_user_tweetcount", # number of tweets sent by the user (at collection time)
    "from_user_followercount", # number of users following the author (at collection time)
    "from_user_friendcount", # number of users the author is following (at collection time)
    "from_user_favourites_count", # number of likes the author has expressed (at collection time)
    "from_user_listed",   # number of users lists the author has been included in (at collection time)
    "from_user_withheld_scope", # whether the user content is withheld, ignorable
    "from_user_withheld_countries", # list of ISO country codes in which the user content is withheld, separated by |, ignorable
    "from_user_created_at", # ISO datetime of creation of the author's account
    "collected_via_thread", # whether the tweet was retrieved only as part of a thread including a tweet matching the desired query
    "retweeted_id",      # digital ID of the retweeted message
    "retweeted_user_name", # text ID of the user who authored the retweeted message
    "retweeted_user_id",  # digital ID of the user who authored the retweeted message
    "quoted_id",         # digital ID of the retweeted message
    "quoted_user_name",  # text ID of the user who authored the retweeted message
    "quoted_user_id",    # digital ID of the user who authored the retweeted message
    "links",              # list of links included in the text content, with redirections resolved, separated by |
    "medias_urls",        # list of links to images/videos embedded, separated by |
    "medias_files",       # list of filenames of images/videos embedded and downloaded, separated by |, ignorable when medias collections isn't enabled
    "mentioned_user_names", # list of text IDs of users mentionned, separated by |
    "mentioned_user_ids", # list of digital IDs of users mentionned, separated by |
    "hashtags"           # list of hashtags used, lowercased, separated by |
]
```

Explorer les dynamiques temporelles



Explorer des réseaux de co-citation de sites web



Graphe obtenu à partir d'un corpus de 60 millions de tweets citant des médias français

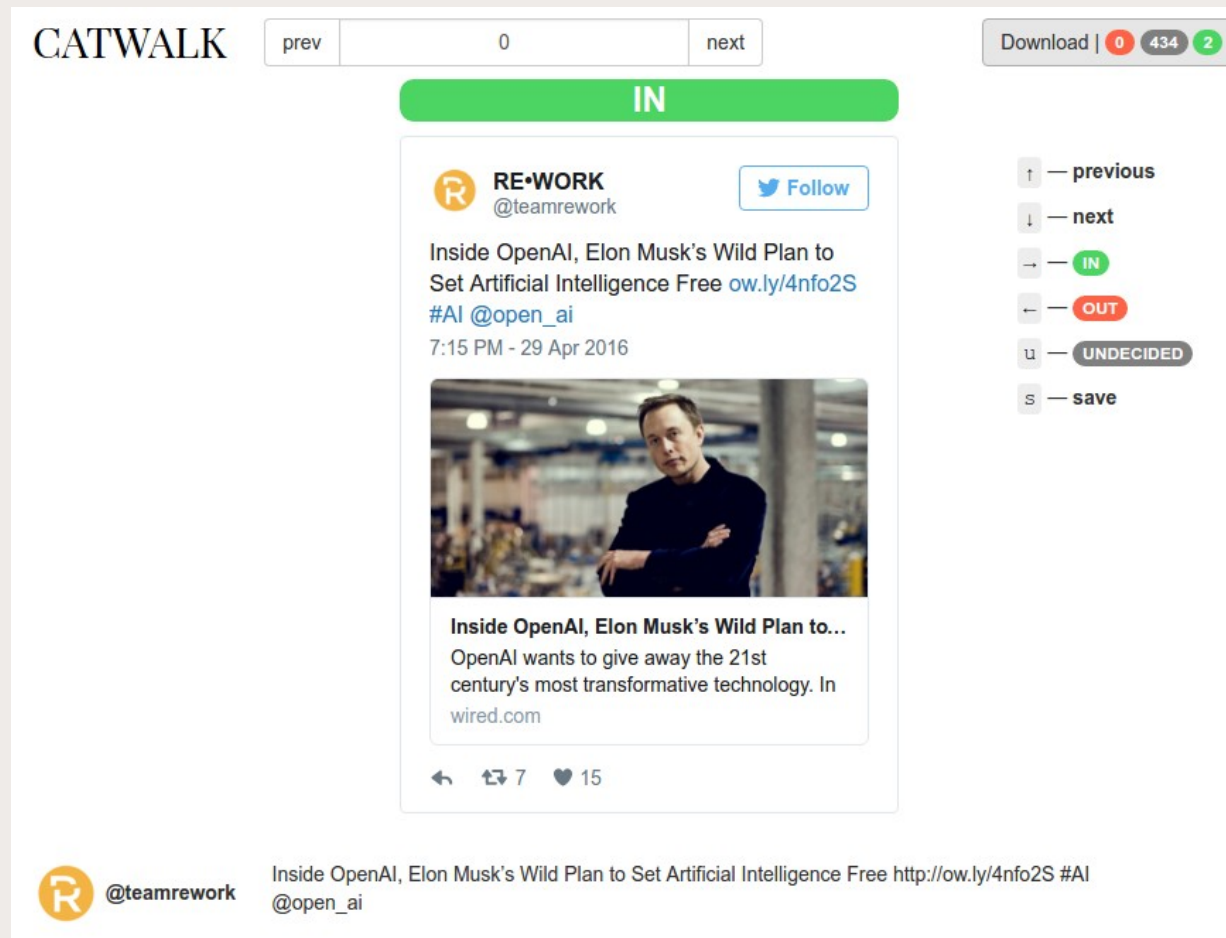
Explorer l'espace visuel d'un corpus



CatWalk : sélection qualitative de tweets

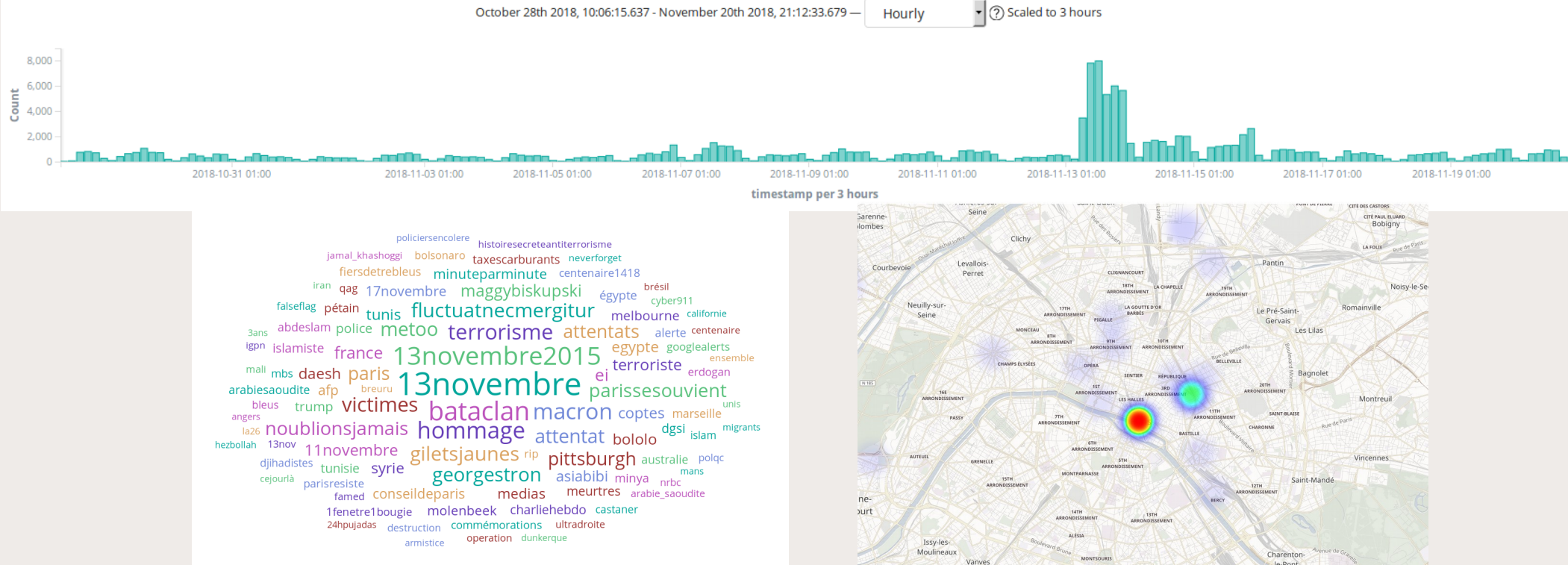
<https://medialab.github.io/catwalk/>

- Passer en revue rapidement « à la Tinder » tous les tweets d'un CSV pour décider de les inclure / exclure d'un corpus



Perspectives futures

- **Achever la migration MongoDB → ElasticSearch (+ Kibana)**



- Développement d'interfaces web pour Gazouilloire
 - Administration des collectes
 - Intégration native de CatWalk
 - Analyse exploratoire visuelle

Merci, et maintenant, à vous de jouer !

<https://github.com/medialab/gazouilloire>



Questions?

benjamin.ooghe@sciencespo.fr


[@boogheta](#) [@medialab_ScPo](#)


Obtenir des clés d'API Twitter


<https://developer.twitter.com/en/apps>


 Developer Use cases Products Docs More Apply Apps 


STATUS: IN PROGRESS

 User profile

 Account details

 Use case details

 Terms of service

 Email verification


Interested in a developer account?

Some of our premium APIs are currently in Beta. By applying, you agree to receive emails from our team requesting feedback on your experience.

Select a user profile to associate

By default, this @username will be the admin of this developer account. If you are creating a developer account on behalf of your organization, you may wish to use your organization's @username as it is most likely to own the Apps you will use to access the API endpoints or warrant special permissions. You'll be able to invite teammates and re-assign roles later within your developer account settings.

Associate your current Twitter @username

 **boogheta**
@boogheta

The phone number associated with this Twitter @username is not verified. You must add a valid phone number and verify it prior to applying for developer access.

Add a valid phone number

[Sign in as a different Twitter @username](#)
[Create new Twitter @username](#)

Installer Gazouilloire et son environnement

(Attention : conçu pour Linux et Mac OS X)

- Installer la base de données MongoDB :
<https://docs.mongodb.com/manual/installation/>

```
# Clone Gazouilloire's git repository (branch elasticPy3)
git clone https://github.com/medialab/gazouilloire.git -b elasticPy3
cd gazouilloire

# Create a Python environment (using PyEnv)
curl https://pyenv.run | bash
pyenv install 3.6.5
pyenv virtualenv 3.6.5 gazouilloire
pyenv activate gazouilloire
pip install -r requirements.txt

# Configure Gazouilloire
cp config.json{.example,}
```

Configurer Gazouilloire

```
{
  "twitter": {
    "key": "xxxxxxxxxxxxxxxxxxxxxx",
    "secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "oauth_token": "xxxxxxxx-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "oauth_secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  },
  "database": {
    "type": "mongo",
    "host": "localhost",
    "port": 27017,
    "db": "gazouilloire"
  },
  "keywords": [
    "giletsjaunes",
    "@DonaldTrump"
  ],
  "url_pieces": [
    "lemonde.fr"
  ],
  "time_limited_keywords": {
    "bieber": [
      ["2014-05-08 16:00", "2014-05-08 16:05"]
    ]
  },
  "language": "fr",
  "geolocalisation": "Paris, France",
  "geolocalisation_type": "city",
  "resolve_redirected_links": true,
  "grab_conversations": true,
  "download_medias": false,
  "medias_directory": "medias",
  "timezone": "Europe/Paris",
  "debug": false
}
```

Lancer une collecte et exporter les données

```
[18:31](elasticPy3 %)=boogheta@medialaptop3:~/dev/gazouilloire-elastic $ ./gazouilloire/run.py
[2019-04-12 18:31:44.361844] INFO: Pile length: 0
[2019-04-12 18:31:44.363171] INFO: Starting stream track until None
[2019-04-12 18:31:44.363317] DEBUG: Calling stream with args {'filter_level': 'none', 'stall_warnings': 'true', 'language': 'fr', 'track': 'giletsjaunes,gilets jaunes,lemonde fr'}
[2019-04-12 18:31:44.739441] INFO: Starting search queries cycle with 606 remaining calls for the next 860 seconds
[2019-04-12 18:31:44.739500] DEBUG: Starting search query on giletsjaunes OR gilets%20jaunes OR url:"lemonde fr" since 0
[2019-04-12 18:31:45.194446] DEBUG: [search] +85 tweets (giletsjaunes OR gilets%20jaunes OR url:"lemonde fr")
[2019-04-12 18:31:45.317005] DEBUG: [stream] +1 tweet
[2019-04-12 18:31:45.662340] DEBUG: [search] +100 tweets (giletsjaunes OR gilets%20jaunes OR url:"lemonde fr")
[2019-04-12 18:31:45.694745] DEBUG: [stream] +1 tweet
[2019-04-12 18:31:45.949036] DEBUG: [stream] +1 tweet
[2019-04-12 18:31:46.155623] DEBUG: [search] +93 tweets (giletsjaunes OR gilets%20jaunes OR url:"lemonde fr")
[2019-04-12 18:31:46.363529] INFO: Pile length: 281
[2019-04-12 18:31:46.408398] DEBUG: [stream] +1 tweet
[2019-04-12 18:31:46.626098] DEBUG: Saved 281 tweets in database
```

```
[18:34](elasticPy3 %)=boogheta@medialaptop3:~/dev/gazouilloire-elastic $ PYTHONPATH=. bin/export_csv_as_tcat.py > tweets.csv
25% (15145 of 58439) |####| Elapsed Time: 0:00:04 ETA: 0:00:13
```

```
# To export a csv with most fields (formatted similarly to [DMI's TCAT](https://github.com/digitalmethodsinitiative/dmi-tcat)):
PYTHONPATH=. bin/export_csv_as_tcat.py

# To export a csv of all tweets having a specific word in their text:
PYTHONPATH=. bin/export_csv_as_tcat.py medialab

# To export a csv of all tweets having one of many specific words in their text:
PYTHONPATH=. bin/export_csv_as_tcat.py medialab digitalhumanities datajournalism '#python'

# To export a csv of all tweets matching a specific MongoDB query, for instance by user_name:
PYTHONPATH=. bin/export_csv_as_tcat.py '{"user_screen_name": "medialab_ScPo"}'

# To export a csv with the most useful fields:
PYTHONPATH=. bin/export_csv.py

# To export the whole text content of the tweets:
PYTHONPATH=. bin/export_all_text.py

# To compute the top shared web domains in the collected tweets:
PYTHONPATH=. bin/export_shared_domains.py
```