Captation de données web

Camille Maussang

camille.maussang@rtgi.fr
RTGI

IC05 - A09

Qui suis-je ?

Qui suis-je ?

- Camille Maussang (`cmaussan`)

Qui suis-je ?

- Camille Maussang (`cmaussan`)
- Chef du dev chez RTGI...

Qui suis-je ?

- Camille Maussang (`cmaussan`)
- Chef du dev chez RTGI...
- ... qui fabrique des outils d'analyse du web social

Qui suis-je ?

- Camille Maussang (`cmaussan`)
- Chef du dev chez RTGI...
- ... qui fabrique des outils d'analyse du web social
- ... en captant des données sur le web ;)

Qu'est-ce que le web et comment le saisir ?

Le web est un corpus de documents

Qu'est-ce que le web et comment le saisir ?

Le web est un corpus de documents

- ouvert,

Qu'est-ce que le web et comment le saisir ?

Le web est un corpus de documents

- ouvert,
- hétérogène,

Qu'est-ce que le web et comment le saisir ?

Le web est un corpus de documents

- ouvert,
- hétérogène,
- et dynamique.

Qu'est-ce que le web et comment le saisir ?

Le web peut être représenté par des graphes

Qu'est-ce que le web et comment le saisir ?

Le web peut être représenté par des graphes

- où les noeuds sont :

Qu'est-ce que le web et comment le saisir ?

Le web peut être représenté par des graphes

- où les noeuds sont :
  - des pages,

Qu'est-ce que le web et comment le saisir ?

Le web peut être représenté par des graphes

- où les noeuds sont :
  - des pages,
  - des sites,

Qu'est-ce que le web et comment le saisir ?

Le web peut être représenté par des graphes

- où les noeuds sont :
  - des pages,
  - des sites,
  - des mots,

Qu'est-ce que le web et comment le saisir ?

Le web peut être représenté par des graphes

- où les noeuds sont :
  - des pages,
  - des sites,
  - des mots,
  - ou des gens,

Qu'est-ce que le web et comment le saisir ?

Le web peut être représenté par des graphes

- où les noeuds sont :
  - des pages,
  - des sites,
  - des mots,
  - ou des gens,
- et les arcs des liens.

Qu'est-ce que le web et comment le saisir ?

Capter des données sur le web requiert un certain nombre de ressources

Qu'est-ce que le web et comment le saisir ?

Capter des données sur le web requiert un certain nombre de ressources

- Bande passante

Qu'est-ce que le web et comment le saisir ?

Capter des données sur le web requiert un certain nombre de ressources

- Bande passante
- Stockage

Qu'est-ce que le web et comment le saisir ?

Capter des données sur le web requiert un certain nombre de ressources

- Bande passante
- Stockage
- Temps machine

Qu'est-ce que le web et comment le saisir ?

Donc :

Qu'est-ce que le web et comment le saisir ?

Donc :

- Que cherchons-nous ?

Qu'est-ce que le web et comment le saisir ?

Donc :

- Que cherchons-nous ?
- Que faire pour récupérer ce qui nous est important ?

Qu'est-ce que le web et comment le saisir ?

Donc :

- Que cherchons-nous ?
- Que faire pour récupérer ce qui nous est important ?
- Toujours penser « heuristiques »...

Qu'est-ce que le web et comment le saisir ?

Donc :

- Que cherchons-nous ?
- Que faire pour récupérer ce qui nous est important ?
- Toujours penser « heuristiques »...
- ... et « effets de bord »!

Qu'est-ce que le web et comment le saisir ?

Ne jamais oublier !

Qu'est-ce que le web et comment le saisir ?

Ne jamais oublier !

Le web c'est

Qu'est-ce que le web et comment le saisir ?

Ne jamais oublier !

Le web c'est n'importe qui (ouvert)

Qu'est-ce que le web et comment le saisir ?

Ne jamais oublier !

Le web c'est n'importe qui (ouvert) qui publie n'importe quoi n'importe comment (hétérogène)

Qu'est-ce que le web et comment le saisir ?

Ne jamais oublier !

Le web c'est n'importe qui (ouvert) qui publie n'importe quoi n'importe comment (hétérogène) n'importe quand (dynamique).

Définitions

Normes, recommandations et standards

Définitions

Normes, recommandations et standards

- Norme (ISO/RFC) : `HTTP`, `URL`, `SGML`, `HTML 1-2`, `MIME`

Définitions

Normes, recommandations et standards

- Norme (ISO/RFC) : `HTTP`, `URL`, `SGML`, `HTML 1-2`, `MIME`
- Recommandation W3C : `HTML 3-4-5`, `XHTML 1`, `CSS`, `DOM`

Définitions

Normes, recommandations et standards

- Norme (ISO/RFC) : `HTTP`, `URL`, `SGML`, `HTML` 1-2, `MIME`
- Recommandation W3C : `HTML` 3-4-5, `XHTML` 1, `CSS`, `DOM`
- Standards : `PDF` et Flash (Taux de pénétration $> 99\%$)

Définitions

Normes, recommandations et standards

- Norme (ISO/RFC) : `HTTP`, `URL`, `SGML`, `HTML` 1-2, `MIME`
- Recommandation W3C : `HTML` 3-4-5, `XHTML` 1, `CSS`, `DOM`
- Standards : `PDF` et Flash (Taux de pénétration $> 99\%$)

Web dynamique

Définitions

Normes, recommandations et standards

- Norme (ISO/RFC) : `HTTP`, `URL`, `SGML`, `HTML 1-2`, `MIME`
- Recommandation W3C : `HTML 3-4-5`, `XHTML 1`, `CSS`, `DOM`
- Standards : `PDF` et Flash (Taux de pénétration $> 99\%$)

Web dynamique

- *server-side* : `CGI`, `PHP`, `ASP`, `JSP`

Définitions

Normes, recommandations et standards

- Norme (ISO/RFC) : `HTTP`, `URL`, `SGML`, `HTML 1-2`, `MIME`
- Recommandation W3C : `HTML 3-4-5`, `XHTML 1`, `CSS`, `DOM`
- Standards : `PDF` et Flash (Taux de pénétration $> 99\%$)

Web dynamique

- *server-side* : `CGI`, `PHP`, `ASP`, `JSP`
- *client-side* : `Javscript`, Flash, ActiveX

Prologue

Principe

Prologue

Principe

- Télécharger *une* page

Prologue

Principe

- Télécharger *une* page

```
$ wget 'http://www.example.org/' -O page.html
$ curl 'http://www.example.org/' > page.html
$ perl -MLWP::Simple -e 'print get("http://www.example.org/")' > page.html
```

Prologue

Principe

- Télécharger *une* page

```
$ wget 'http://www.example.org/' -O page.html
$ curl 'http://www.example.org/' > page.html
$ perl -MLWP::Simple -e 'print get("http://www.example.org/")' > page.html
```

Déjà des problèmes

Prologue

Principe

- Télécharger *une* page

```
$ wget 'http://www.example.org/' -O page.html
$ curl 'http://www.example.org/' > page.html
$ perl -MLWP::Simple -e 'print get("http://www.example.org/")' > page.html
```

Déjà des problèmes

- Type de fichier

Prologue

Principe

- Télécharger *une* page

```
$ wget 'http://www.example.org/' -O page.html
$ curl 'http://www.example.org/' > page.html
$ perl -MLWP::Simple -e 'print get("http://www.example.org/")' > page.html
```

Déjà des problèmes

- Type de fichier
- Encodage

Prologue

Principe

- Télécharger *une* page

```
$ wget 'http://www.example.org/' -O page.html
$ curl 'http://www.example.org/' > page.html
$ perl -MLWP::Simple -e 'print get("http://www.example.org/")' > page.html
```

Déjà des problèmes

- Type de fichier
- Encodage
- Contenu (HTML)

Crawler

Principe

Crawler

Principe

- Télécharger 1 page

Crawler

Principe

- Télécharger 1 page
- Extraire les liens

Crawler

Principe

- Télécharger 1 page
- Extraire les liens
- Télécharger les pages pointées par les liens

Crawler

Principe

- Télécharger 1 page
- Extraire les liens
- Télécharger les pages pointées par les liens
- etc. etc.

Crawler

```perl
use strict; use warnings;
use LWP::Simple;

my ( $max_depth, @seed ) = @ARGV or die( 'need depth and url(s)' );
my @already_visited = ();
my $depth = 0;
my @to_visit = @seed;

while( $depth <= $max_depth && @to_visit ) {
    print "crawling depth $depth\n";
    my @links = ();
    for my $url ( @to_visit ) {
        if( my $content = get( $url ) ) {
            while ( $content =~ m/<a href="([^"]+)"/gi) { push @links, $1 }
        }
        push @already_visited, $url;
        print "$url visited.\n";
    }
    @to_visit = ();
    for my $url_to_check ( @links ) {
        my $to_push = 0;
        for my $url_visited ( @already_visited ) {
            if( $url_to_check eq $url_visited ) { $to_push = 0; last; }
            $to_push = 1;
        }
        push @to_visit, $url_to_check
            if( $to_push && !grep{ $_ eq $url_to_check } @to_visit );
    }
    $depth++;
}
print "end.\n";
```

Crawler

- Métriques (distance, profondeur, etc.)

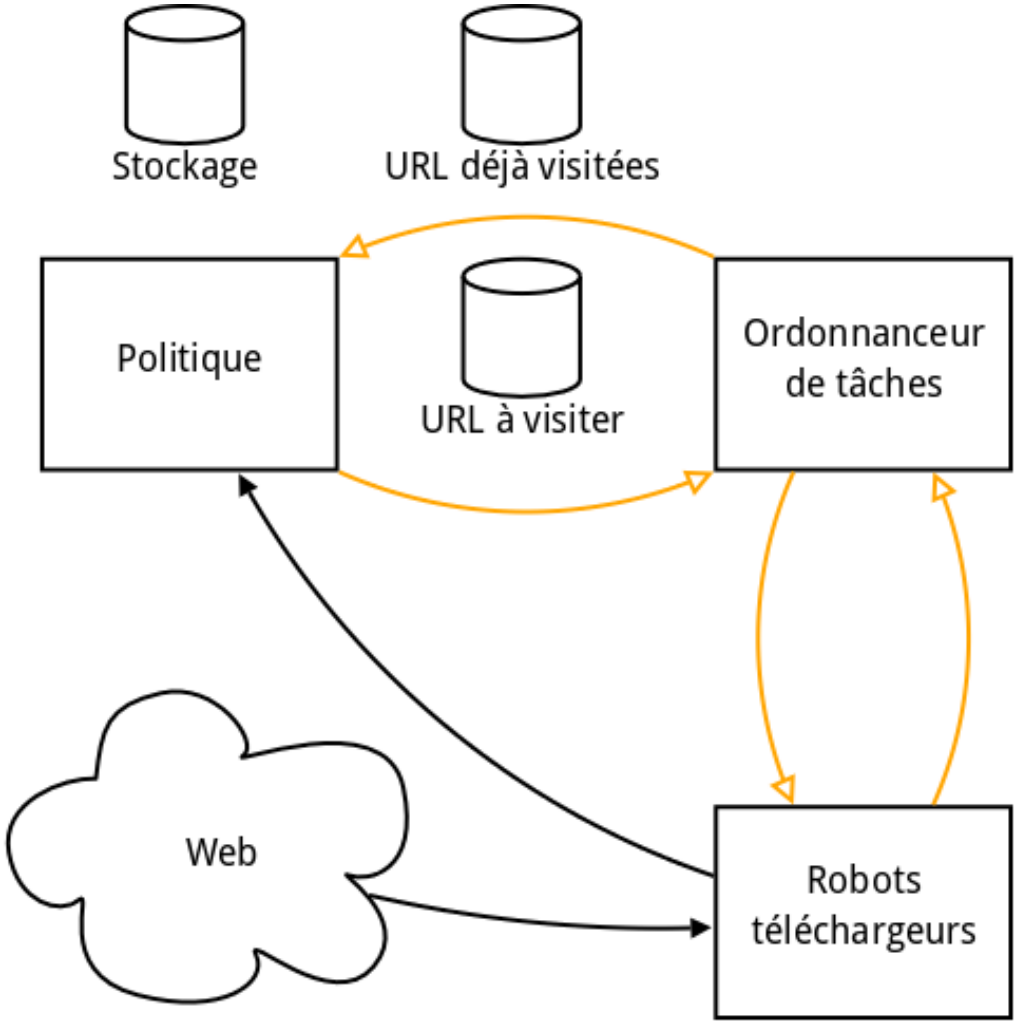
Crawler

- Métriques (distance, profondeur, etc.)
- Performance (goulots d'étranglement)

Crawler

- Métriques (distance, profondeur, etc.)
- Performance (goulots d'étranglement)
- Scalabilité (de 1 page à 1G pages)

Crawler


Stockage
URL déjà visitées
Politique
URL à visiter
Ordonnanceur de tâches
Web
Robots téléchargeurs

Crawler

De nouveaux problèmes

Crawler

De nouveaux problèmes

- Politesse

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)
- Addressage

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)
- Addressage
  - Normalisation d'`URL` (doublons)

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)
- Addressage
  - Normalisation d'`URL` (doublons)
  - Site ou page ?

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)
- Addressage
  - Normalisation d'`URL` (doublons)
  - Site ou page ?
  - Plusieurs permaliens pour un seul contenu (GYM aide un peu)

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)
- Addressage
  - Normalisation d'`URL` (doublons)
  - Site ou page ?
  - Plusieurs permaliens pour un seul contenu (GYM aide un peu)
- Autres...

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)
- Addressage
  - Normalisation d'`URL` (doublons)
  - Site ou page ?
  - Plusieurs permaliens pour un seul contenu (GYM aide un peu)
- Autres...
  - Javascript

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)
- Addressage
  - Normalisation d'`URL` (doublons)
  - Site ou page ?
  - Plusieurs permaliens pour un seul contenu (GYM aide un peu)
- Autres...
  - Javascript
  - *Deep web*

Crawler

De nouveaux problèmes

- Politesse
  - DoS (*Denial of Service*) : DNS, Serveurs HTTP
  - Blacklistage officiel (`robots.txt`, `sitemap.xml`, etc.)
  - Blacklistage officieux (*cloaking*, pièges à robot)
- Addressage
  - Normalisation d'`URL` (doublons)
  - Site ou page ?
  - Plusieurs permaliens pour un seul contenu (GYM aide un peu)
- Autres...
  - Javascript
  - *Deep web*
  - Web privé

Crawler

Astuces

Crawler

Astuces

- Utiliser les `headers HTTP`

Crawler

Astuces

- Utiliser les `headers HTTP`
- User-agent

Crawler

Astuces

- Utiliser les `headers HTTP`
- User-agent
- `random et sleep`

Crawler

Astuces

- Utiliser les `headers HTTP`
- User-agent
- `random` et `sleep`
- Multi-agent plutôt que multi-thread

Crawler

Astuces

- Utiliser les `headers HTTP`
- User-agent
- `random` et `sleep`
- Multi-agent plutôt que multi-thread

Principes du *Focused crawler*

Crawler

Astuces

- Utiliser les `headers HTTP`
- User-agent
- `random` et `sleep`
- Multi-agent plutôt que multi-thread

Principes du *Focused crawler*

- Ne télécharger que les pages pertinentes

Crawler

Astuces

- Utiliser les `headers HTTP`
- User-agent
- `random` et `sleep`
- Multi-agent plutôt que multi-thread

Principes du *Focused crawler*

- Ne télécharger que les pages pertinentes
- Indicateurs topologiques

Crawler

Astuces

- Utiliser les `headers HTTP`
- User-agent
- `random` et `sleep`
- Multi-agent plutôt que multi-thread

Principes du *Focused crawler*

- Ne télécharger que les pages pertinentes
- Indicateurs topologiques
- Indicateurs sémantiques

Aggrégation

Principe

Aggrégation

Principe

Syndication ou comment *renverser* l'accès aux données

Aggrégation

Principe

Syndication ou comment *renverser* l'accès aux données

Problèmes

Aggrégation

Principe

Syndication ou comment *renverser* l'accès aux données

Problèmes

- Atom, RSS, encore mille versions

Aggrégation

Principe

Syndication ou comment *renverser* l'accès aux données

Problèmes

- Atom, RSS, encore mille versions
- Flux complet / partiel / vide ...

Aggrégation

Principe

Syndication ou comment *renverser* l'accès aux données

Problèmes

- Atom, RSS, encore mille versions
- Flux complet / partiel / vide ...
- ... avec ou sans date, permaliens, HTML

Scraping

Principe

Scraping

Principe

Analyser une page web pour en extraire une information spécifique

Scraping

Principe

Analyser une page web pour en extraire une information spécifique

Problèmes

Scraping

Principe

Analyser une page web pour en extraire une information spécifique

Problèmes

- DOM ou Regexp ou les deux

Scraping

Principe

Analyser une page web pour en extraire une information spécifique

Problèmes

- `DOM` ou Regexp ou les deux
- Template et dynamisme des pages scrapées

Scraping

Principe

Analyser une page web pour en extraire une information spécifique

Problèmes

- `DOM` ou Regexp ou les deux
- Template et dynamisme des pages scrapées
- Flash et Javascript

API

Principe

API

Principe

Utiliser les API de certains sites pour collecter la donnée

API

Principe

Utiliser les API de certains sites pour collecter la donnée

Problèmes

API

Principe

Utiliser les API de certains sites pour collecter la donnée

Problèmes

- Limitations

API

Principe

Utiliser les API de certains sites pour collecter la donnée

Problèmes

- Limitations
- API propriétaires

Un exemple concret

Créer un corpus de documents sur un thème précis avec Google

Un exemple concret

Créer un corpus de documents sur un thème précis avec Google

- Créer un set de requêtes

Un exemple concret

Créer un corpus de documents sur un thème précis avec Google

- Créer un set de requêtes
- Écrire un robot de captation

Un exemple concret

Créer un corpus de documents sur un thème précis avec Google

- Créer un set de requêtes
- Écrire un robot de captation
  - Module de scraping des résultats de Google (avec `Web::Scraper` par ex.)

Un exemple concret

Créer un corpus de documents sur un thème précis avec Google

- Créer un set de requêtes
- Écrire un robot de captation
  - Module de scraping des résultats de Google (avec `Web::Scraper` par ex.)
  - Module d'ordonnancement

Un exemple concret

Créer un corpus de documents sur un thème précis avec Google

- Créer un set de requêtes
- Écrire un robot de captation
  - Module de scraping des résultats de Google (avec `Web::Scraper` par ex.)
  - Module d'ordonnancement
  - Module de « crawl »

Un exemple concret

Créer un corpus de documents sur un thème précis avec Google

- Créer un set de requêtes
- Écrire un robot de captation
  - Module de scraping des résultats de Google (avec `Web::Scraper` par ex.)
  - Module d'ordonnancement
  - Module de « crawl »
- Capter les données :)

Wikipédia est ton ami :)

- http://en.wikipedia.org/wiki/HTML
- http://en.wikipedia.org/wiki/Web_crawler
- http://en.wikipedia.org/wiki/Focused_crawler
- http://en.wikipedia.org/wiki/Web_scraping
- http://en.wikipedia.org/wiki/URL_normalization
- http://en.wikipedia.org/wiki/Cloaking
- http://en.wikipedia.org/wiki/User_agent
- http://en.wikipedia.org/wiki/Spider_trap
- http://en.wikipedia.org/wiki/Denial-of-service_attack
- etc.

Merci !

- http://labs.rtgi.eu/
- http://github.com/cmaussan/Picrowler
- http://github.com/cmaussan/captation-ic05-a09-tex

Merci !

- http://labs.rtgi.eu/
- http://github.com/cmaussan/Picrowler
- http://github.com/cmaussan/captation-ic05-a09-tex