GNUSIDE

Aide à la définition d'un projet pour le concours Mendeley
Compte-rendu de consultation

Deuxième Labo - Gnuside

Vendredi 8 juillet 2011

1 Questions sur Mendeley

1.1 Définitions

API : Application Programming Interface. Ici il s'agit d'un ensemble de fonctions retournant des résultat issus du site Mendeley, mise à disposition au public via un ensemble d'URL.

Quota : La mise à disposition de l'API de Mendeley au public se fait sous un ensemble de conditions légale et techniques. Entre autres une limite de connexion par heure au dela de laquelle l'application se retrouve bloquée.

Whitelist : Liste des applications autorisées, dans une certaine limite à dépasser le quota initial sur l'API.

Blacklist : Liste des applications bloquées et n'ayant plus accès à l'API.

1.2 Pour chaque tag Mendeley, est-il possible d'afficher ses *related tags* ?

Note : Comme, par exemple, sur `http://www.mendeley.com/tags/evolution/`

Cette information n'est pas directement accessible via l'API.

Il faudrait chercher, pour chaque tag, les documents associés et les autres tags de ceux-ci (ce qui constitue l'ensemble des related tags). Attention cependant au nombre de connexions sur l'API pour ce faire.

Un autre méthode serait de faire du scrapping, c'est à dire récupérer la page HTML de Mendeley sur laquelle cette information s'affiche et l'en extraire (avec une bibliothèque comme *Mechanize* en Ruby, ou similaire selon les choix techniques du projet).

Pour cette méthode, qui d'un point de vue technique fonctionne très bien, les connexions ne sont pas comptabilisée dans les quotas de l'API mais elle est interdite dans les conditions d'utilisation du site. L'idéal serait de prévenir l'équipe de Mendeley pour passer dans la liste-blanche pour l'API et seulement au cas échéant négocier avec eux pour faire du scrapping en attendant une meilleure API.

1.3 Combien de tags y a-t-il au total dans la base Mendeley ?

Le nombre de tags et la liste des tags ne sont pas disponibles avec l'API actuelle.

Il n'y a pas de pistes "efficace" pour obtenir ces informations mais il reste possible de prendre un un dictionnaire (éventuellement spécifique à un domaine) et tester l'existance du mot en tant que tag sur la page sur Mendeley.

Selon l'objectif on peut également utiliser seulement un échantillon de ce dictionnaire.

GNUSIDE

1.4 Combien de tags correspondent à un article Wikipedia EN ?

Note : on considère ici les articles ayant le même nom qu'un tag donné.

Pour savoir combien de tags correspondent à un article Wikipedia, il suffit de compter le nombre de tags, qui répondent positivement à test d'existance de page (voir en 2.1 pour plus de détails).

Il serait cependant préférable d'avoir la base Wikipedia en local pour limiter le nombre de hits sur l'API si l'on compte tester l'existance de chaque tag (si l'on parle de la liste complète comme précédemment).

1.5 Peut-on obtenir la longueur de chacun de ces articles...

Note : Il s'agit de ranger les tags par longueur de l'article décroissante, puis tracer le graphique : longueur de la page Wikipedia EN = f(rang du tag) ? On s'attend a une loi de Zipf, qui ressortira encore mieux dans un repère log-log.

Pour un tag donné, obtenir la longueur de l'article correspondant dans wikipedia est très facile. Il faut commencer par obtenir le contenu de l'article comme indiqué en 2.2.

Une fois le contenu de la page obtenu, calculer sa longueur et trier selon celle-ci est trivial.

Reste à tracer un graphique, à l'aide de l'outil *gnuplot* par exemple ou d'une bibliothèque adaptée aux choix techniques du projet.

1.6 Faire la même chose avec la longueur de la page de discussion de l'article Wikipedia EN

Sur Wikipedia, la discussion concernant un article porte le même nom que cet article avec le préfixe `Talk:`.

Pour en obtenir le contenu (et la longueur) de l'article $TITRE$ il suffit donc d'utiliser l'URL : `http://en.wikipedia.org/w/api.php?action=query&prop=revisions&titles=Talk:TITRE&rvprop=content&format=xml`

1.7 Peut-on faire des recherches sur les utilisateurs ?

Il n'y a pas de recherche sur les utilisateurs prévue dans l'API de Mendeley. Par contre pour un groupe donné on peut savoir qui en est l'auteur et qui y participe.

1.8 Peut-on lier des tags à des disciplines ?

On ne peut pas directement lier des tags à des disciplines via l'API.

Par contre c'est possible en plusieurs requêtes :

$$Tag \Rightarrow [Document_1, ..., Document_n]$$

$$Document_i \Rightarrow [Discipline_1, ..., Discipline_{n'}]$$

Par transitivité, on peut obtenir :

$$Tag \Rightarrow [Discipline_1, ..., Discipline_{n''}]$$

Au coût de $n' + 1$ connexions sur l'API.

1.9 Rechercher le nombre d'articles associés à des tags ?

Pour trouver le nombre d'articles associés au tag TAG il suffit de compter le nombre de résultats d'une requête sur l'API, avec l'url : `http://api.mendeley.com/oapi/documents/tagged/TAG/`

Exemple :


```
{
    "total_results": 1249,
    "total_pages": 63,
    "current_page": 0,
    "items_per_page": 20,
    "documents": [
        {
            "id": 280936,
            "title": "Service-oriented science.",
            "publication_outlet": "Science New York NY",
            "year": 2005,
            "mendeley_url": "http:\/\/www.mendeley.com\/research\/serviceoriented-science\/",
            "doi": null,
            "authors": "Foster."
        },
        .... [lots of documents descriptions]....
    ]
}
```


1.10 Récupérer la popularité des tags par semaine ?

Note : Cette information se trouve sur le site : `http://www.mendeley.com/stats/` tout en bas de la page.

Récupérer la popularité des tags via l'API semble possible mais il faut passer par chacune des disciplines. Par exemple pour le tag TAG sur l'adresse : `http://api.mendeley.com/oapi/documents/tagged/TAG/`

Exemple :

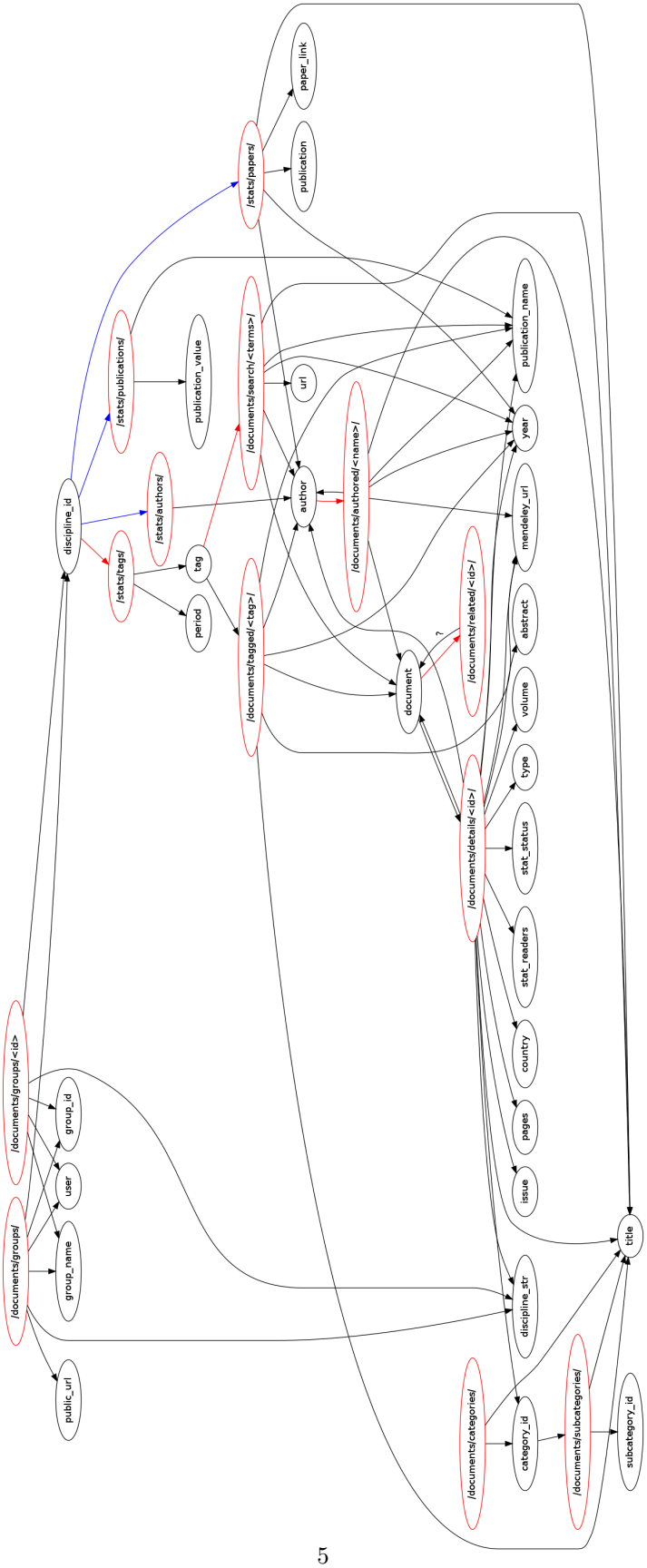

```
[
    {
        "period": "15-21 May 2010",
        "tags": "No tags for this period."
    },
    {
        "period": "22-28 May 2010",
        "tags": "No tags for this period."
    },
    {
        "period": "29 May - 4 June 2010",
        "tags": "No tags for this period."
    }
]
```


1.11 Récupérer les groupes liés à des tags ?

Cette information est disponible mais par via l'API.

Il semble nécessaire, pour un tag TAG, ici de faire du scrapping sur l'url `http://www.mendeley.com/groups/tagged/TAG/` .

1.12 Graphe de l'API Mendeley (simplifié)


/documents/groups/<id>
/documents/groups/
public_url
group_id
user
group_name
discipline_id
/stats/publications/
publication_value
/stats/authors/
/stats/tags/
tag
period
/stats/papers/
paper_link
publication
/documents/search/<terms>/
uri
author
/documents/authored/<name>/
/documents/tagged/<tag>/
document
?
/documents/related/<id>/
/documents/details/<id>/
publication_name
year
mendeley_url
abstract
volume
type
stat_status
stat_readers
country
pages
issue
title
discipline_str
/documents/categories/
category_id
/documents/subcategories/
subcategory_id

1.13 Autres remarques

Nombre de connexions

Pour limiter le nombre de connexions, il faudra certainement mettre en cache local (fichiers textes brut ou en base de données) tous les résultat obtenus, autant sur l'API que sur les pages de Mendeley. De plus

API Proxy

De plus, selon les contraintes de l'application à développer, il peut être pratique de développer une sorte de proxy à l'API de Mendeley, avec des méthodes en plus pour ce qui nécessiterait aujourd'hui du scrapping. Ainsi le jour où, du fait d'une probable l'évolution de l'API Mendeley, ce scrapping ne serait plus nécessaire, il y aurait uniquement le proxy à « corriger ».

OAuth

Contrairement à ce qui est indiqué dans la documentation de l'API Mendeley, c'est le protocole OAuth2 qu'il faut utiliser et non pas OAuth...

2 Questions sur Wikipedia

2.1 Savoir si une page Wikipedia existe ?

Pour tester l'existance d'une page Wikipedia, on utilisera l'URL suivante : `http://en.wikipedia.org/w/api.php?action=query&titles=DOESNOTEXIST&format=xml`

Le résultat sera de la forme :

```
<?xml version="1.0"?>
<api>
    <query>
        <pages>
            <page ns="0" title="DOESNOTEXIST" missing="" />
            <page pageid="600799" ns="0" title="FOO" />
            <page pageid="11178" ns="0" title="Foobar" />
        </pages>
    </query>
</api>
```

Si le *label* missing de la balise *page* est rempli, c'est que la page n'existe pas.

2.2 Obtenir le contenu d'une page Wikipedia ?

Pour obtenir le contenu d'une page $TITRE$ sur Wikipedia il faut accéder à l'URL suivante : `http://en.wikipedia.org/w/api.php?action=query&prop=revisions&titles=TITRE&rvprop=content&format=xml` .

Le resultat ressemblera à :

```
<?xml version="1.0"?>
<api><query><pages>
    <page pageid="8410" ns="0" title="TITRE">
        <revisions>
            <rev xml:space="preserve">
            ....[page content].....
            </rev>
        </revisions>
    </page>
</api>
```

Le contenu de la page, à la révision courante, se situe entre les balises *rev*.

2.3 Est-ce possible d'identifier les articles scientifiques cités en référence ou dans « further reading » en général ? Sous quelle forme ?

Il n'y a pas de requête permettant d'accéder directement à la section « further reader » d'un article.
Cependant il est facile d'extraire cette section une fois le contenu de l'article téléchargé.

2.4 Est-ce qu'on peut savoir si une entrée a été supprimée, fondue, redirigée ou désambigüée ?

Oui, cette information est indiquée soit dans le texte, soit dans les catégories de l'article. Il faut simplement en télécharger le contenu.

2.5 Peut-on connaître le rating d'une entrée ?

Si par *rating* on comprends le nombre d'apparitions d'une entrée TAG dans les autres articles de Wikipedia, alors on emploiera l'URL suivante de l'API :`http://en.wikipedia.org/w/api.php?action=query&list=search&srsearch=TAG&srprop=score`.
Le résultat sera quelque chose comme :

```
<?xml version="1.0"?>
<api>
    <query>
        <searchinfo totalhits="1150" />
        <search>
            <p ns="0" title="TAG" />
            <p ns="0" title="TAG (magazine)" />
            <p ns="0" title="TAG (company)" />
            <p ns="0" title="TAG (KDE)" />
            <p ns="0" title="Chamber (comics)" />
            <p ns="0" title="TB" />
            <p ns="0" title="TAG watt" />
            <p ns="0" title="Doctor TAG" />
            <p ns="0" title="TAG Festival" />
            <p ns="0" title="TAG (disambiguation)" />
        </search>
    </query>
    <query-continue>
        <search sroffset="10" />
    </query-continue>
</api>
```

Le *rating* y sera la valeur de l'attribut *totalhits* dans le tag *searchinfo*.

2.6 La(les) catégorie(s) ou portail(s) dont fait partie l'article ?

Il existe une fonction de l'API Wikipedia qui offre cette possibilité. Pour une page $TITRE$ donnée, il s'agit de l'URL `http://en.wikipedia.org/w/api.php?action=query&prop=categories&titles=TITRE`, qui retournera un résultat de la forme :

```
<?xml version="1.0"?>
<api>
    <query>
        <pages><page pageid="736" ns="0" title="TITRE">
            <categories>
                <cl ns="14" title="Category:1879 births" />
                <cl ns="14" title="Category:1955 deaths" />
                <cl ns="14" title="Category:19th-century German people" />
                <cl ns="14" title="Category:Academics of the Charles University" />
                <cl ns="14" title="Category:All articles with dead external links" />
                <cl ns="14" title="Category:All articles with unsourced statements" />
                <cl ns="14" title="Category:All pages needing cleanup" />
                <cl ns="14" title="Category:American Jews" />
               <cl ns="14" title="Category:American humanitarians" />
            </categories>
        </page></pages>
    </query>
    <query-continue>
        <categories clcontinue="736|American pacifists" />
    </query-continue>
</api>
```

2.7 Toutes les statistiques d'une entrée ?

Note : (voir page http://toolserver.org/~soxred93/articleinfo/index.php?article=Reproductive_health&lang=en&wiki=wikipedia par exemple)

Les statistiques de wikipedia sont accessibles sur l'API à l'exception du compteur de vues qui a été desactivé en raison d'un trop grand nombre de visites.

L'information est disponible à travers des sites tiers comme `http://stats.grok.se/` qui analysent les logs anonymisés de Wikipedia(accessibles sur `http://dammit.lt/wikistats`).

Pour un tag TAG donné, les données brutes sont téléchargeables à l'adresse `http://toolserver.org/~emw/index.php?c=rawdata&m=get_traffic_data&p1=TAG&project1=en&from=12/10/2007&to=7/11/2011`

Pour plus d'informations, voir `http://en.wikipedia.org/wiki/Wikipedia:Statistics`

2.8 Peut-on récupérer tous les mots correspondant à des entrées wikipedia dans le texte d'un article ?

On peut récupérer les mots correspondant à des entrés wikipédia soit en analysant la syntaxe Wiki de l'article téléchargé, soit via l'API, pour l'article $TITRE$ à l'URL suivante : `http://en.wikipedia.org/w/api.php?action=query&prop=links&titles=TITLE`

2.9 Peut-on récupérer les entrées situées dans la section « see also » ?

Oui, il est possible de récupérer les entrées de la section « see also », ce de la même façon que la section « further reader » (voir en 2.3).

2.10 Autres remarques

Comme pour Mendeley, pour limiter le nombre de connexions et améliorer les performances de l'application, il faudra certainement mettre en cache local tous les résultats obtenus lors des connexions sur l'API de Wikipedia.

Selon le nombre de connexions et les traitements envisagées sur l'API de Wikipedia, il peut être intéressant d'utiliser Wikipedia en local. En effet, les archives de la base de données de l'encyclopédie sont téléchargeables en plusieurs formats, que chacun peut installer, utiliser et traiter selon ses propres contraintes.

Attention cependant au fait que toutes les pages (dicussions, etc.) ne sont pas inclues dans tous les formats d'archives. À titre indicatif l'archive « légère » la plus récente (articles EN uniquement) utilise approximativement 6 Gio sur le disque. Il faudra donc probablement trouver un compromis entre temps d'accès et volume de données à stocker localement.

Pour plus d'informations à ce sujet, consulter les pages `http://en.wikipedia.org/wiki/Wikipedia:Database_download` et `http://dumps.wikimedia.org/enwiki/`.

3 Questions sur Wikio

Remarque : Aucune documentation sur l'API de Wikio n'est disponible publiquement. Il est cependant possible de faire du scrapping sur les pages accessibles aux utilisateurs, comme évoqué comme éventualité pour les sites précédents.

Les questions concernant Wikio, posées dans la préparation de la séance de travail, sont tout de même indiquées ci-après.

3.1 Il me semble qu'il serait intéressant de pouvoir faire des recherches sur les catégories...

Note : Ce que je comprends c'est les « related » quand on fait une recherche sont des noms de catégories en fait. Elles forment une arborescence classique.

De plus, ces « related » sont, me semble-t-il liés seulement à la catégorie et non à la recherche particulière qu'on a faite.

- *...quand une catégorie a-t-elle été créée ?*
- *...récupérer les sous-catégories et la surcatégorie d'une catégorie donnée*

3.2 Est-ce possible de construire un nuage de tags à partir de toutes les catégories auxquelles un article serait lié ?

Notes : Lors d'une recherche, sous chaque article apparaît un « Explore » avec les catégories auxquelles cet article est reliée.

Pour peu qu'une liste de tags ou de catégories soit d'abord récupérée, il est aisé de construire un nuage de tags.

3.3 Est-ce possible de faire des recherches sur une échelle de temps qu'on choisirait ?

Notes : Sachant qu'une recherche me semble fournir des résultats très récents (quelques jours).

3.4 Est-il possible de faire une recherche en limitant le scope à une ou plusieurs catégories ?

3.5 Comment se présentent les résultats de recherche ?

4 Pistes pour la candidature au concours Mendeley

On se propose de réaliser un outil permettant d'aider à la catégorisation de documents.

L'outil prendra en paramètre un tag T_0 et retournera un ensemble de tags $\{T, ..., T_n\}$ choisis selon les critères suivants.

4.1 Association audacieuse

Approche naïve :

- Chercher les papiers associés au tag entré ;
- Classer les papiers du plus au moins populaire (en fonction du nombre de readers) ;
- Proposer les tags qui apparaissent le moins souvent ensembles.

Approche avancée :

- Chercher les papiers associés au tag entré
- Classer les papiers du plus au moins populaire (en fonction du nombre de readers)
- Proposer les associations de tags qui apparaissent le moins souvent ensembles parmis les tags les plus populaires
  Nota-Bene : pour cette derniere étape, il faudra permettre de choisir la proportion de tags à considérer parmi les tags les plus populaires.

4.2 Association controversée

Pistes :

- Articles fréquemment édités sur Wikipedia ;
- Articles avec de grandes ou « lourdes » éditions sur Wikipedia ;
- Articles brouillons ;
- On définit le rang des articles : $ControversyRank(T) = (nombre\,d'éditions(T))/(nombre\,de\,visites(T))$;
- Catégories spéciales d'articles dues aux *weasel-words*.

Approche naïve :

- Lister les articles Wikipedia ou les deux mots T_0 et W apparaissent ;
  Nota-Bene : il faudra permettre de choisir seulement une proportion de tags à considérer dans cette liste (peut-être seulement les x premiers) ;
  Remarque : il manque une étape. Il faudrait définir d'où vient W. Est-ce des related tags sur Mendeley, des liens de la page T_0 de Wikipedia, d'autre chose ?
- Trier les articles T_i en fonction de $ControversyRank(T_i)$.
- Proposer les T_i les plus controversés.

4.3 Association qui suscite l'intérêt - sujets « chauds »

Pistes :

- Tags qui sont co-utilisés dans des groupes de Mendeley
- Tags *related* dont la dérivée de la fonction de popularité est positive sur les x derniers mois ou années (parmi plusieurs tags, on choisira celui dont la dérivée de la popularité a la pente la plus raide).

Approche naïve :

- Lister les *related tags*
- Trier par nombre d'occurences et par rang (en nombre d'apparition sur wikipedia)
- Proposer ceux qui ont le rang le plus proche de celui du tag courant.

4.4 Association extra-disciplinaire

Soit $RelatedTags(T) = \{$ l'ensemble des related tags de$T$ $\}$
et $Categories(T) = \{$ les catégories associées à la page T dans Wikipedia$\}$
et $CatDistinct(T) = \{Categories(T) \setminus Categories(T_0)\}$

et $CategoryWeight(C) = Card(\{TR_i \mid \exists TR_i \in RelatedTags(T_0), C \in CatDistinct(TR_i)\})$

Note : c'est à dire le nombre d'apparitions de C dans l'ensemble des categories distinctes pour chacun des related tags *de T_0*

et enfin

$$TagWeight(TR_i) = \frac{\sum_{C_i \in Categories(TR_i)} CategoryWeight(C_i)}{Card(Categories(TR_i))}$$

Remarque : est-il plus judicieux de diviser par $Card(Categories(TR_i))$ ou bien $Card(CatDistinct(TR_i))$?

Approche naïve :

- Pour tout $TR_i \in RelatedTags(T_0)$ faire
  - Trier les catégories par nombre d'apparition dans l'ensemble des TR_i (on obtient un poids par catégorie) ;
  - Pour chaque tag TR_i , calculer son poids :
  - Trier les tags TR_i selon leur $TagWeight(TR_i)$;
  - Proposer ceux qui ont les poids les plus faibles.

-