Re-création d'une table unifiée par revues

notes ISTEX-RD nov 2015 (R. Loth)

1) Objectifs

Les références bibliographiques utilisent le plus souvent une forme abrégée des titres, ce qui est gênant pour valider la résolution de la bib dans notre API.

1.1) Dans le cadre de la résolution

- résolution des refbibs extraites => liste des abréviations sera très utile
  - les auteurs des refbibs utilisent l'abréviation qui leur paraît la plus "courante" pour chaque revue
  - ex:
    - L'AUTEUR A MIS : **"J. Mol. Biol."** (et c'est extrait par grobid)
    - ON DOIT RETROUVER DANS L'API: **"Journal of Molecular Biology"**
      - NB: les jokers lucene comme `q=host.title:(j* mol* biol*)` donneraient ici le bon résultat
      - mais pour "Natl" => "national" la stratégie par jokers ne marche pas
      - et surtout on veut *valider* le résultat de façon régulière
  - solution: extrait de Millenium avec ces mêmes infos
- ces abrév. suivent aussi "mot par mot" la norme ISO4 et son extension dite LTWA à l'ISSN Portal
  - cette table est non bijective
    - "Physics" => "phys."
    - "Physik" => "phys."
    - "Physica" => "phys."
    - "Specialized" => "spec."
    - "Special" => "spec."
  - dans les refbibs à résoudre, on trouve des variantes autour de ce standard
  - envisageable :
    - récupération de la table mot pat mot (crawling possible)
    - validation avec les alternatives
      - ex: "spec." matche "Special|Specialized"

1.2) Autres roblématiques liées

D'autres tâches d'ISTEX-RD et ISTEX-DATA peuvent bénéficier d'une table enrichie des revues:

- filtrer ce qu'on cherche par période => liste de présence de la revue dans l'API
  - généré simplement par requête "agrégations" sur le serveur elastic avec la version raw de host.title et une aggreg "max(publicationDate)"
- autres usages:
  - corriger les rares notices ayant un champ "host.title" erroné
    - ex: 7092FB412DFD580B2B68FC7F43604DC78EF6D940
    - ce doc serait apparu si on listait tous les host.title de l'API, en regardant les "revues" ne comportant **qu'un seul doc** <=> coquille dans le titre
  - reporter des catégories WOS => liste par revues

Le travail pour lister les abréviations dans le script table_hosttitle.sh peut

servir pour relever PAR ISSN d'autres choix de facettes que host.title.

Une éventuelle table par ISSN qui en résulterait permettrait d'avoir une meilleure vue d'ensemble sur les revues.

Plus cette table serait riche, plus on aurait la maîtrise de nos métadonnées (correction notices, décision si on a un doc dans la base, stats d'ensemble)

"Riche" veut ici dire qu'on peut joindre à chaque ISSN présent chez nous le plus de métadonnées supplémentaires (nb de docs dans ISTEX, dates mini/maxi, abréviations, catégories...)

2) Liste initiale des revues présentes dans ISTEX

Rendu facile par les améliorations aux "facettes" de l'équipe API courant 2015

```
# récupération de la facette
curl 'https://api.istex.fr/document/?q=*&facet=host.issn\[*\]&size=1'> facette_issn.json

# conversion du json en une table
jq -r '.aggregations["host.issn"].buckets[].key' facette_issn.json > issn
jq -r '.aggregations["host.issn"].buckets[].docCount' facette_issn.json > ndocs

paste issn ndocs > table.tsv

# on peut ouvrir la table dans excel ou calc
soffice -calc table.tsv
```

> NB: on peut faire le même traitement pour host.title via l'API. On pourrait aussi faire la combinaison des deux via elastic directement (cf. https://github.com/elastic/elasticsearch/issues/5100 et notamment la solution de jpountz avec un mapping copy_to) mais on va la faire à l'étape (4.1) plus loin par un script https://git.istex.fr/loth/utilitaires/blob/master/table_hosttitle.sh

3) Table extraite de Millenium

3.1) Pour info: autres pistes actuellement non suivies

https://images.webofknowledge.com/WOK46/help/WOS/A_abrvjt.html

- Liste très large datant de 2008
  - contient ce qu'on veut mais contient aussi des entrées "fourre-tout"
  - ex: ACID RAIN RESEARCH: DO WE HAVE ENOUGH ANSWERS? <=> **STUD ENVIRON SCI**
  - ici ce n'est pas le titre de la revue "Studies in Environmental Science" mais celui d'un volume
- Pas d'ISSNs
- Les images du WOK des années suivantes ne contiennent plus la liste

http://journalseek.net/ et http://cassi.cas.org/

- Efficaces et couverture large
- On peut faire des requêtes mais la **table** n'est pas accessible
  - pour journalseek elle est vendue à part par OCLC

autres liens sur:

- lien ancien: http://www2.iastate.edu/~cyberstacks/JAS.htm
- très intéressant: http://www.library.illinois.edu/biotech/j-abbrev.html

3.2) Table de Millenium

Quantitativement, Millenium enregistre + de 53000 notices "Series" (= par revues)

Parmi celles-ci il y en a ~ 30000 comportant des abbréviations dans le champ 531 : on se cantonne à ces notices pour les décomptes de couverture.

Titre Millenium 200	ISSN Millenium 011	Abbrév Millenium 531|a
Abdominal imaging	0942-8925	Abdom. imaging
Acta alimentaria polonica	0137-1495	Acta Aliment. Pol.
	(...)	
JAMA. The journal of the American Medical A	0098-7484	JAMA j. Am. Med. Assoc.
	(...)	
Journal of modern optics	0950-0340	J. mod. opt.
Journal of molecular and applied genetics	0271-6801	J. mol. appl. genet.
Journal of molecular biology	0022-2836	J. mol. biol.
	(...)	
Zoophysiology and Ecology	0084-5663	Zoophysiol. Ecol.
Zuckerindustrie;(Berlin, West)	0344-8657	Zuckerindustrie

Extrait simplifié de la table Millenium *par Series*
en vert les cas qui correspondent directement à notre besoin
en orange les cas à recodages potentiellement nécessaire

Points forts:

- les abréviations sont directement au format courant qu'on recherche
  - entrées avec soin durant des années par les agents du catalogage
  - c'est le même format que celui utilisé par les auteurs des refbibs
  - ces abréviations sont encore maintenues et actualisées à l'INIST (nov 2015)
- la couverture est de 80% pour les revues ISTEX
  - sans aucun traitement, on retrouve 13M de nos documents dans les infos millenium-series via les ISSN (sur les 16M)
  - cela correspond à 3590 revues retrouvées parmi nos 6300 ISSN,
    - on retrouve notamment toutes nos principales (à gros volumes de docs)

Points faibles

- Certains cas rares où une abréviation est tellement courante qu'elle a supplanté le nom de revue (ex typique : "JAMA") et où par convention, nos documentalistes l'ont inclue au titre, ce qui n'est pas la pratique suivie dans l'API
- Certains cas rares où le lieu de publication est inclus dans le titre (pratique courante par le passé pour désambiguïser des revues "synonymes", mais non suivie dans ISTEX)

Conclusion

On utilise donc la base extraite par Volker Stock de Millenium pour valider les résolutions de refbibs extraites à titre abrégé => docs de l'API à titre entier.

4) Intégration

4.1) Jonction et complétion des infos via des tables

La première opération consiste pour chaque ISSN de l'API à reprendre son titre exact dans l'API, puis on va y ajouter les infos des abréviations.

Pour cela je reprends la liste des ISSNs faite en (2) et je fais le script table_hosttitle.sh (il utilise la facette host.title de l'API sur chaque ISSN pour ramener le ou les titre(s) associés) (la liste d'ISSN doit être passée sans quotes autour des ISSN !!)

/!\ si l'API a un titre différent du titre plein habituel recensé dans Millenium, c'est le titre API qu'on va garder

Table informative complète

On fait alors la jonction des deux tables ce qui nous donne une table avec les colonnes suivantes:
ISSN | api_host.title| api_docCount | abréviation

NB : l'ISSN n'est pas unique à ce stade car il peut y avoir plusieurs host.title par ISSN

Cette table intermédiaire est disponible sous doc/infos_revues_jonction-2015.ods

Dans la colonne abréviation, si on se limite aux titres ayant plus de 3000 documents, il manque une trentaine d'abréviations millenium, on va les chercher manuellement sur le site http://journalseek.net/
Pour les titres groupant moins de 3000 docs dans l'API au 24/11/2015, on ne complète pas si on n'a pas déjà l'info.

Normalisation des abréviations

On utilise la commande suivante pour stocker les abréviations sans leurs ponctuation et majuscules, car elles ne sont pas régulières dans les sources

```
# ponctuation => espaces
tr -s "[:punct:]" " " < abrevs \
# réduction des espaces multiples
| tr -s " " \
# suppression d'espaces en début/fin de chaîne
| sed 's/^ \+//; s/ \+$//' \
# passage en minuscules
| tr '[:upper:]' '[:lower:]' > abrevs_sans_punct
```

Table compacte à intégrer

La table informative complète est un peu lourde pour l'intégration.
On gardera pour le programme resolver.py uniquement la colonne des ISSN et des abréviations normalisées présentes.

Cette table prête pour l'intégration est disponible sous etc/issn_abrevs.tsv

4.2) Intégration de la table au script `resolver.py`

Resolver fonctionne autour de 2 étapes clés:

étape 1 => interrogation de l'API
- objectif : avoir le plus grand **rappel**, même avec des faux positifs)
- type de requête lancée : série de fragments *champ:(mot1 mot2)*
- fonctions dédiées dans `resolver.py`:
  - `BiblStruct.bib_subvalues()`
  - `BiblStruct.prepare_query_frags()`
  - `get_top_match_or_None`

étape 2 => validation par des règles de comparaison intelligentes
- objectif : avoir une **précision** parfaite tout en gardant le plus de résultats
- fonctions dédiées dans `resolver.py`:
  - `BiblStruct.test_hit()`
    - qui fait des tests sur des infos nécessaires et suffisantes à valider
    - et utilise dans chaque test une fonction de comparaison de 2 chaînes de cara : `soft_compare(str_extrait_PDF, str_réponse_API)`
- l'évaluation test_findout du sprint RD 5 avait montré que l'abréviation était l'élément le plus difficile à valider

La table créée s'intègre donc à l'étape 2.

On modifie la fonction `test_hit()` de l'objet `BiblStruct`.
- pour permettre de lancer la comparaison `soft_compare()` aussi bien sur la version extraite du PDF que sur sa traduction en titre non-abrégé
- dans ce cas, on valide si et seulement si l'ISSN de notre table est identique à l'ISSN du doc pointé dans l'API