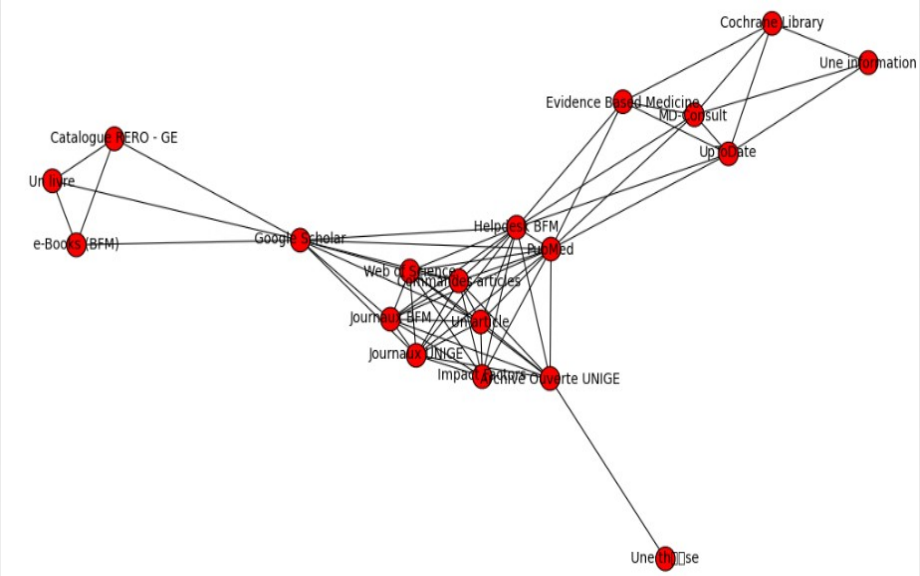
MarcXimiL



Analyse de similarité

- **détection de doublons**
- **détection préprint/article**
- **bibliométrie**
- **suggestions à l'usager**
- structure de collections
- veille documentaire
- détection de plagiat

http://marcximil.sourceforge.net

MarcXimiL : caractéristiques

- Fonctionne avec tout logiciel compatible MARCXML.
  - Invenio (CERN, INSPIRE, RERODOC, Infoscience)
  - Filtre d'importation pour Virtua
- Standards utilisés pour la compatibilité (MARCXML, OAI-PMH2, XML-RPC, UTF8)
- Multi-plateforme (Python 2.4.0 → 3.1.1 [dernière version])
- Algorithmes de recherche d'informations (et autres)
- Open source : GPLv3
- **Flexibilité dans l'analyse: stratégies de similarité**

MarcXimiL : configuration flexible

Flexibilité dans la définition de stratégies d'analyse de similarité:

- **Chargement** (MARCXML, OAI-PMHv2)
- **Extraction** des notices: quels champs, comment les combiner (concaténer [ex: titre+soustitre], ensembles[100$a+700$a])
- **Normalisation** (case, diacritiques, ponctuation) et **indexation** des champs (identique, ntf-nidf, shingling, digrams, etc.). Option: utiliser MySQL (plus rapide).
- **Schéma des comparaisons** : 1coll, 2colls, pré-clustering [futur: v. 0.3.3]
- **Similarité entre les champs:** auteurs, date/année, doi/uid, textuelles (vectorielle [Dice,Salton,Jaccard, Shingles], probabiliste [OKAPI-BM25], orthographique [Levenshtein], ensembliste [Initials, Digrams]), ...
- **Similarité entre notices** : moyennes pondérées (classiques & rapides), et fonctions spécialisées (maxsim, ubiquist, boundaries, ...)
- **Seuil de rapport** (sur la similarité globale)

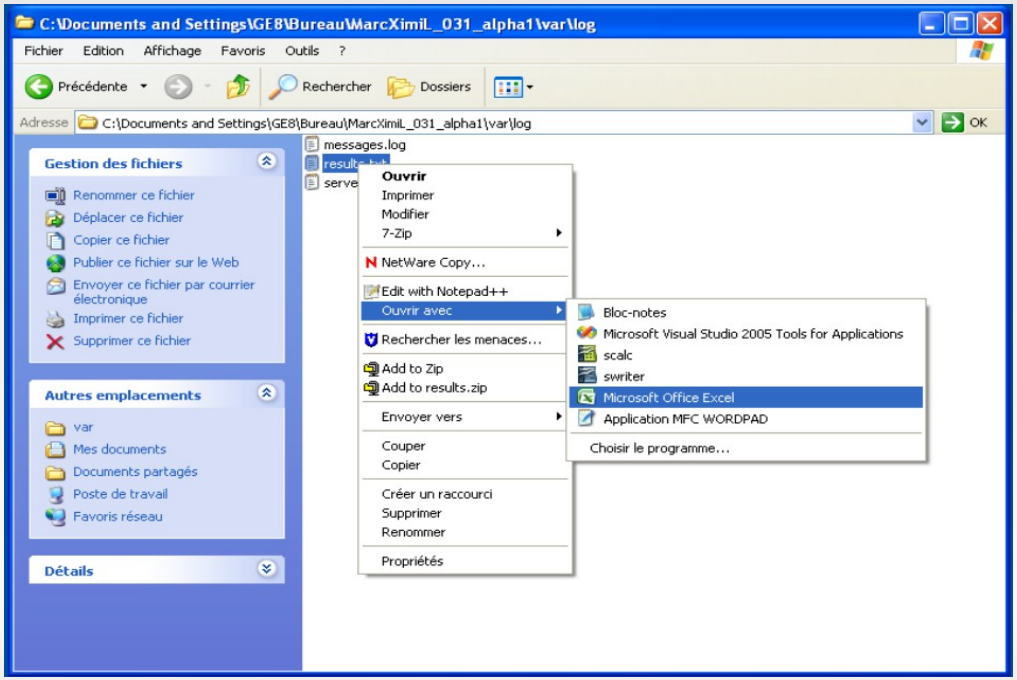
Utilisation : ligne de commande

Appel en ligne de commande (ouvrir un termial, répertoire ./bin)

1) Charger: `python similarity.py -l coll -f marc.xml -s similarity_config__digrams`
2) Comparer: `python similarity.py -c coll -o results.txt -s similarity_config__digrams`
3) MàJ: `python similarity.py -u coll -f update.xml -s similarity_config__digrams`
4) Tuer: `python similarity.py -k coll`

NB: nombreuses autres options disponibles, et utilisation via l'API.

Les résultats peuvent être exprimés en XML (machine à machine), ou tabulés (par défaut).



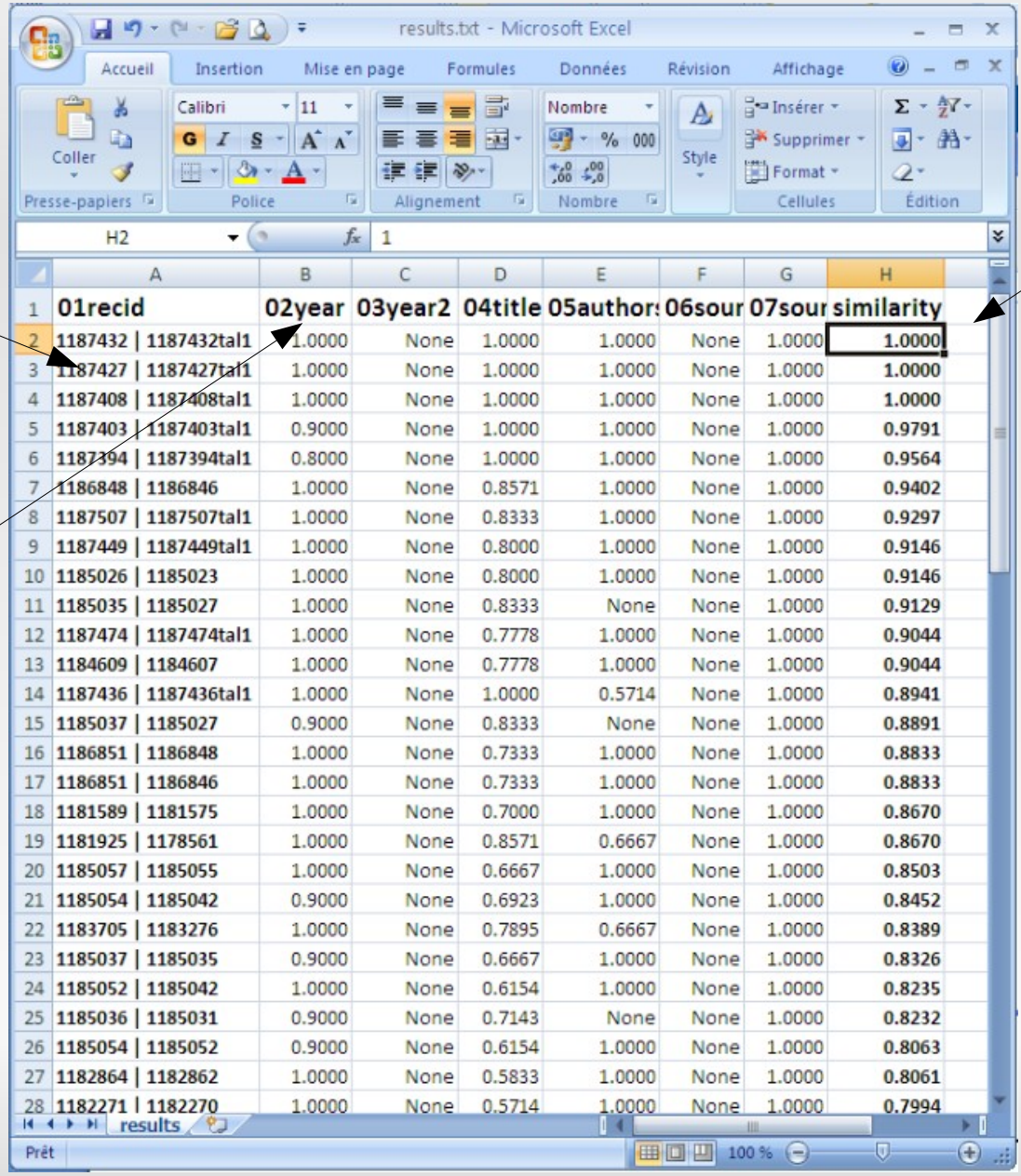
Résultats

Les identifiants des notices comparées sont transmis dans les résultats.

Les valeurs de similarité entre les champs sont renseignées.

None est renvoyé si pas applicable.

Par défaut: tri par similarité globale décroissante



	A	B	C	D	E	F	G	H
1	01recid	02year	03year2	04title	05author	06sour	07sour	similarity
2	1187432 | 1187432tal1	1.0000	None	1.0000	1.0000	None	1.0000	1.0000
3	1187427 | 1187427tal1	1.0000	None	1.0000	1.0000	None	1.0000	1.0000
4	1187408 | 1187408tal1	1.0000	None	1.0000	1.0000	None	1.0000	1.0000
5	1187403 | 1187403tal1	0.9000	None	1.0000	1.0000	None	1.0000	0.9791
6	1187394 | 1187394tal1	0.8000	None	1.0000	1.0000	None	1.0000	0.9564
7	1186848 | 1186846	1.0000	None	0.8571	1.0000	None	1.0000	0.9402
8	1187507 | 1187507tal1	1.0000	None	0.8333	1.0000	None	1.0000	0.9297
9	1187449 | 1187449tal1	1.0000	None	0.8000	1.0000	None	1.0000	0.9146
10	1185026 | 1185023	1.0000	None	0.8000	1.0000	None	1.0000	0.9146
11	1185035 | 1185027	1.0000	None	0.8333	None	None	1.0000	0.9129
12	1187474 | 1187474tal1	1.0000	None	0.7778	1.0000	None	1.0000	0.9044
13	1184609 | 1184607	1.0000	None	0.7778	1.0000	None	1.0000	0.9044
14	1187436 | 1187436tal1	1.0000	None	1.0000	0.5714	None	1.0000	0.8941
15	1185037 | 1185027	0.9000	None	0.8333	None	None	1.0000	0.8891
16	1186851 | 1186848	1.0000	None	0.7333	1.0000	None	1.0000	0.8833
17	1186851 | 1186846	1.0000	None	0.7333	1.0000	None	1.0000	0.8833
18	1181589 | 1181575	1.0000	None	0.7000	1.0000	None	1.0000	0.8670
19	1181925 | 1178561	1.0000	None	0.8571	0.6667	None	1.0000	0.8670
20	1185057 | 1185055	1.0000	None	0.6667	1.0000	None	1.0000	0.8503
21	1185054 | 1185042	0.9000	None	0.6923	1.0000	None	1.0000	0.8452
22	1183705 | 1183276	1.0000	None	0.7895	0.6667	None	1.0000	0.8389
23	1185037 | 1185035	0.9000	None	0.6667	1.0000	None	1.0000	0.8326
24	1185052 | 1185042	1.0000	None	0.6154	1.0000	None	1.0000	0.8235
25	1185036 | 1185031	0.9000	None	0.7143	None	None	1.0000	0.8232
26	1185054 | 1185052	0.9000	None	0.6154	1.0000	None	1.0000	0.8063
27	1182864 | 1182862	1.0000	None	0.5833	1.0000	None	1.0000	0.8061
28	1182271 | 1182270	1.0000	None	0.5714	1.0000	None	1.0000	0.7994

Résultats: précision (version 0.3.2)

Méthodologie:

- Test en **double aveugle**:
  - 8 collections de 2000 notices dont 10 quasi-doublons.
  - Total: 16'000 notices dont 80 quasi-doublons.
- Sources métadonnées: **CERN, ETHZ, RERODOC**
- Quasi-doublons de test: erreurs fréquents de catalogage, règles de catalogage, translittération, formats (html vs. text), variation systématiques (au sein et sur le nombre de champs), etc.

STRATEGY	Recall 10	Recall 20	Recall 50
ubiquist	71.3%	93.8%	96.3%
digrams	61.3%	83.8%	92.5%
Combinés :	~95%		

Résultas: timing (digrams, version 0.3.2)

Une collection

- Toute une collection de 5'000 notices: 7 min 40
- Toute une collection de 10'000 notices: 1 h 10

Deux collections

- Flux entrant de 100 notices par semaine comparées avec les notices des 2 dernières années (~10'000): -> 1min20
- Idem pour 1000 notices par semaine (2 ans: ~100'000): -> 2h20

Live (API)

- 1 notice avec une collection de 10'000 : ~ 1 s

NB: performance sur ordinateur de bureau UNIGE.

Futur v0.3.3 : pré-clustering & gestion des résultats

1 - Gestion des résultats [pas commencé]

- Actuellement, les résultats de chaque analyse sont stockés dans un fichier.
- Améliorer: stockage SQL, présentation (tri, limites), croisement des résultats, caching (API), etc.

2 - Accélération par pré-clustering [en cours]:

Les notices sont passées en revue une par une ... Pour chaque notice, comparaison seulement des notices susceptibles d'être des doublons selon le pré-analyse (pour un champ donné qui contiennent au moins un des deux mots dont les DF indiquent qu'ils sont les plus pertinents).

Résultats préliminaires (sans les temps de chargement, sur SQLite):

- 2000 notices -> ~1 min 30 , accélération ~ 4x
- ~ 17'000 notices -> 32 min , accélération ~ 12x

MarcXimiL : conclusion

- Outil efficace de détection de quasi-doublons.
- Atout principal : très grande flexibilité.
- Adapté à l'archive ouverte de l'UNIGE?
  - Utilisation : à côté, sur ordinateur de bureau (?)
  - Vitesse : aucun problème.
  - Détection : aucun problème.

Pour s'en assurer : analyse d'une/de collection/s test basée/s sur les méta-données de l'archive ouverte avec doublons fabriqués.