Rapport Projet Biblio Scolaire

Eté 2023

André Diler

Contents

[Mise en situation 3](#_Toc144159185)

[Problématique 3](#_Toc144159186)

[Comment aligner ? 4](#_Toc144159187)

[Evolution données du graphe de connaissances 4](#_Toc144159188)

[Données fin juillet 4](#_Toc144159189)

[Données manquantes 4](#_Toc144159190)

[Doublons 5](#_Toc144159191)

[Données 4 aout 6](#_Toc144159192)

[Données manquantes 6](#_Toc144159193)

[Doublons 6](#_Toc144159194)

[Données 8 aout 7](#_Toc144159195)

[Données manquantes 7](#_Toc144159196)

[Doublons 7](#_Toc144159197)

[Méthode alignement 7](#_Toc144159198)

[Métriques 7](#_Toc144159199)

[Création de clé 8](#_Toc144159200)

[Exact 9](#_Toc144159201)

[Approximatif 9](#_Toc144159202)

[Complexité 9](#_Toc144159203)

[Parallélisation 9](#_Toc144159204)

[Améliorations 10](#_Toc144159205)

[Hybride 10](#_Toc144159206)

[Collisions 11](#_Toc144159207)

[Format fichiers alignements 11](#_Toc144159208)

[Colonnes CSV 11](#_Toc144159209)

[Champs RDF 11](#_Toc144159210)

[Alignements BNF & Constellation 12](#_Toc144159211)

[Résultats 12](#_Toc144159212)

[Conclusions 12](#_Toc144159213)

[Alignements BNF, Constellation & Lurelu 12](#_Toc144159214)

[Resultats 12](#_Toc144159215)

[Analyse 13](#_Toc144159216)

[Erreur(s) ISBN Lurelu 13](#_Toc144159217)

[A-t-on besoin de l’alignement hybride si on a les ISBN ? 13](#_Toc144159218)

[Organisation des fichiers 14](#_Toc144159219)

[Code source 14](#_Toc144159220)

[Données 15](#_Toc144159221)

# Mise en situation

Un professeur veut trouver un roman psychologique pour sa classe d’adolescents du secondaire.

Il veut que le livre ait des prix ainsi que de bonnes critiques.

Trouver le bon livre lui prend demande beaucoup de temps et de connaissances : il faut bien connaitre les différents site web qui référencent les livres jeunesse francophones, et chaque site web a des attributs spécifiques.

A book cover with a painting of a person walking

Description automatically generatedOn obtient plusieurs candidats de livres avec Constellation avec un filtre « romans psychologiques ».

A yellow and red airplane

Description automatically generated with medium confidenceA person's head with a transparent umbrella

Description automatically generated

Mais il n’y a pas les avis critiques que l’on cherche.

On doit les chercher sur BNF, Lurelu et Babelio, sachant que Lurelu n’offre pas d’options de recherche.

Doit également chercher sur d’autres sites comme Ricochet pour avoir une liste complète des prix littéraires.

## Problématique

Comment simplifier sa tache ?

* Créer graphe de connaissances à partir de différentes sources
* Aligner ces graphes

# Comment aligner ?

Intuitivement, on aligne les livres par ISBN car cette identifiant est unique pour chaque ouvrage. Néanmoins, certaines sources comme Lurelu ne le mentionne pas, donc on doit créer notre propre identifiant unique.

# Evolution données du graphe de connaissances

Karim nettoyait et enrichissait les données pendant que j’en extrayais des connaissances. J’ai séparé les données en 3 versions (fin juillet, 4 août et 8 août).

La première version date de fin juillet car c’est la première version avec laquelle je peux faire un alignement grossier entre BNF et Constellation tout en mesurant avec les ISBN (plus de détail sur la performance plus tard).

Rappel des données extraites :

Constellation : Tous les livres

BNF : Tous les livres jeunesse ayant eu des critiques ou coup de cœurs

Lurelu : Tous les livres (ISBN annotés à la main pour certains livres)

## Données fin juillet

Localisation :

[Books-Python\andre\data\data as of end of july\intra stats 2 august](https://github.com/KarimLak/Books-Python/blob/b8ad74c9055a8ef8b476d733052f9ad746e6ceb9/andre/data/data%20as%20of%20end%20of%20july/intra%20stats%202%20august)

L’ISBN dans BNF et Constellation permettent de mesurer la performance.

Beaucoup de doublons BNF car les livres qui ont des coups de cœurs ont généralement des critiques (donc extraites 2 fois par Karim).

Nombre de livres qui ont des données manquantes pour chaque source (aussi appelé IntraDBStats dans le code)

### Données manquantes

|  |  |  |
| --- | --- | --- |
| Données manquantes / Source | BNF | Constellation |
| Sans ISBN | 341 | 13 |
| Sans nom | 5 | 0 |
| Sans auteur | 1128 | 0 |
| Sans âge | 3283 | 1 |
| Sans date de publication | 95 | 0 |
| Sans éditeur | 45 | 0 |
| Total livres | ~36K parmi ~500K | 11267 |

On peut déjà voir que Constellation a des données de meilleure qualité que BNF.

### Doublons

On regarde aussi les doublons de clés à l’intérieur de chaque source pour voir à quel point notre clé est pertinente et précise. Si on a beaucoup de doublons, la clé n’est pas un bon substitut à l’ISBN.

Plusieurs clés sont testées, car les dates en Constellation et BNF ne correspondent pas (date de 1ère parution extraite au lieu date d’édition dans Constellations), et les éditeurs ne sont pas encore intégrés dans le graph par Karim.

|  |  |  |
| --- | --- | --- |
| Clé / source -> nombre doublons (ou triplets ou plus) | BNF | Constellation |
| ISBN | 5454 | 1 (ISBN Manquants) |
| Nom | 5976 | 241 |
| Nom\_auteur | 5719 | 54 |
| Nom\_auteur\_editeur | 5216 | 13 |
| Nom\_auteur\_editeur\_date | 5266 | 1 |

La plupart des doublons de clé de BNF sont dus aux doublons d’œuvres de BNF expliqués plus haut.

#### Doublons ISBN et analyse des EAN

Tous les doublons ISBN de Constellation sont dus à des ISBN manquants sur le page web.

On peut aussi noter qu’on a de vrais doublons d’ISBN dans de rare cas, comme les coffrets de livres.

Exemple :

<https://catalogue.bnf.fr/ark:/12148/cb43402762j>

<https://catalogue.bnf.fr/ark:/12148/cb43402763w>

Il serait intéressant de déterminer la proportion d’ouvrages qui ont des ISBN identiques, leur raison et leur proportion : un vrai doublon ? Des livres d’un même coffret qui ont le même ISBN ?

Les livres d’un même coffret devraient-ils avoir des ISBN différents ?

De plus, certains ouvrages ont un ISBN et EAN différent. Parfois, on observe plusieurs ouvrages dans une notice :

<https://catalogue.bnf.fr/ark:/12148/cb467610438>

Parfois, l’ISBN est différent de l’EAN. Cela s’explique par l’évolution de la norme ISBN.

<https://catalogue.bnf.fr/ark:/12148/cb361711172>

Il y a une correspondance directe entre l’ISBN-13 et l’EAN-13 : si l’EAN fait 13 charactères, comme l’ISBN, on peut l’aligner à l’ISBN pour augmenter nos chances d’alignement :

De plus, si on a un EAN et un ISBN différent pour le même livre, on peut utiliser ces deux identifiants pour augmenter nos chances d’alignements avec des sources qui utilisent l’une ou l’autre. L’EAN est disponible dans les données de BNF, mais je ne les pas encore vérifiés ni exploitées.

[Quelle est la différence entre ISBN-10 et ISBN-13? – Conseils Rapides](https://conseilsrapides.com/des-articles/quelle-est-la-difference-entre-isbn-10-et-isbn-13/)

Néanmoins, l’ISBN 10 utilisé avant 2007 est différent de l’EAN. Puisque l’EAN est utilisé pour les livres récents (notamment dans Babelio), il serait utile de conserver tous les EAN et ISBN de chaque notice pour augmenter notre alignement.

#### Nombres de doublons par rapport à la précision des clés

On observe que plus la clé est précise, moins on a de doublons. On élimine la clé avec le nom unique du livre car elle n’est pas assez précise, et qu’on ne voudrait probablement pas aligner un livre avec uniquement un nom.

## Données 4 août

[Books-Python\andre\data\data as of 04 august\intra stats as of 4 august](https://github.com/KarimLak/Books-Python/blob/96f7f4664697ff17286183fec63f8480e6ec7e53/andre/data/data%20as%20of%2004%20august/intra%20stats%20as%20of%204%20august)

Plus de doublons BNF : on passe de 36K livres à 30K : les données manquantes et les doublons BNF diminuent naturellement

Erreurs dans Constellation corrigées

* Dates Constellation = date édition
* Titre & auteurs correspondent à page web de Constellation (car avant, extraits directement du csv exporté par Constellation)

### Données manquantes

|  |  |  |
| --- | --- | --- |
| Données manquantes / Source | BNF | Constellation |
| Sans ISBN | 286 | 13 |
| Sans nom | 4 | 0 |
| Sans auteur | 921 | 0 |
| Sans âge | 3068 | 1 |
| Sans date de publication | 74 | 0 |
| Sans éditeur | 38 | 0 |
| Total livres | ~30K parmi ~500K | ~11K |

### Doublons

|  |  |  |
| --- | --- | --- |
| Clé / source -> nombre doublons (ou triplets ou plus) | BNF | Constellation |
| ISBN | 141 | 1 |
| Nom | N/A | N/A |
| Nom\_auteur | 858 | 53 |
| Nom\_auteur\_editeur | 523 | 14 |
| Nom\_auteur\_editeur\_date | 59 | 1 |

## Données 8 aout

[Books-Python\andre\data\data as of 8 august\intra stats as of 8 august](https://github.com/KarimLak/Books-Python/tree/b8ad74c9055a8ef8b476d733052f9ad746e6ceb9/andre/data/data%20as%20of%208%20august/intra%20stats%20as%20of%208%20august)

Les éditeurs sont terminés, et on a moins de données manquantes pour BNF.

### Données manquantes

|  |  |  |
| --- | --- | --- |
| Données manquantes / Source | BNF | Constellation |
| Sans ISBN | 286 | 13 |
| Sans nom | 0 | 0 |
| Sans auteur | 921 | 0 |
| Sans âge | 3068 | 1 |
| Sans date de publication | 32 | 0 |
| Sans éditeur | 9 | 0 |
| Total livres | ~30K parmi ~500K | ~11K |

On peut voir que les dates de publication et éditeurs manquants pour BNF diminuent beaucoup.

Il manque également des dates de publication pour certaines œuvres de Lurelu : « [Le pays des genoux](https://www.lurelu.net/coupsdecoeur03.html)». Vu qu’il y a seulement ~200 œuvres, ça ne vaut peut-être pas le coup de s’attarder sur ces données manquantes puisqu’on peut leur ajouter les ISBN à la main.

### Doublons

|  |  |  |
| --- | --- | --- |
| Clé / source -> nombre doublons (ou triplets ou plus) | BNF | Constellation |
| ISBN | 141 | 13 |
| Nom | N/A | N/A |
| Nom\_auteur | 857 | 53 |
| Nom\_auteur\_editeur | 557 | 13 |
| Nom\_auteur\_editeur\_date | 70 | 1 |

Les doublons augmentent légèrement pour BNF lorsqu’on inclut l’éditeur. Cela peut s’expliquer par le nettoyage des éditeurs par Karim, qui réunit un même éditeur écrit différemment sur des notices différentes.

# Méthode alignement

## Métriques

On mesure la qualité de l’alignement entre deux ouvrages en comparant leur ISBN. Le but est de tester notre algorithme d’alignement (même si on pourrait aligner les 2 ouvrages sans erreurs avec leurs ISBN), et voir s’il est généralisable.

Grace à cette méthode, on peut créer une matrice de confusion :

VP : ISBN des deux ouvrages **alignés** sont les **mêmes**

FP : ISBN des deux ouvrages **alignés** sont les **différents**

VN : un ouvrage **non-aligné** **n’aurait pas pu être aligné** car le même ISBN est **inexistant** dans l’autre source

FN : un ouvrage **non-aligné** **aurait pas pu être aligné** car le même ISBN **existe** dans l’autre source

Ces métriques sont implémentées dans : [interdbstats.InterDbStats.compute\_alignment\_confusion\_matrix\_validation](https://github.com/KarimLak/Books-Python/blob/96f7f4664697ff17286183fec63f8480e6ec7e53/andre/interdbstats.py#L81)

## Création de clé

Au début, la clé se composait des différents attributs du livre tel que récupéré dans le graphe de connaissance, séparés par des « \_ ».

Ensuite, des prétraitements ont été appliqués sur les attributs individuels, avant qu’ils ne soient assemblés (voir le fichier [utils.py](https://github.com/KarimLak/Books-Python/blob/96f7f4664697ff17286183fec63f8480e6ec7e53/andre/utils.py) )

Exemple avec clé nom\_auteur\_date :

Et si on parlait de politique?\_Jeanne Boyer\_\_2014

etsionparlaitdepolitique\_jeanneboyer\_\_2014

N.B. Même si je n’avais pas fait de statistiques sur les doublons de nom\_auteur\_date, j’ai utilisé cette clé pour tester des alignements avec les données du 4 aout car Karim venait de corriger les dates d’édition de Constellation, et elles correspondaient aux dates d’édition de BNF. Les éditeurs n’étant pas encore terminés, la clé la plus précise et qui maximisait l’alignement entre BNF et Constellation était celle-ci.

Plusieurs méthodes de prétraitements ont été effectué (sur les données du 4 aout), et des mesures de performances ont été faites pour chaque méthode pour prouver son utilité.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| modification clé n\_a\_d  Alignement exact BNF & Constellation | originale | sans espace | sans espace  & majuscule | sans espace  & majuscule & accent | sans espace  & majuscule & accent & char spéciaux |
| alignements P | 1136 | 1222 | 1288 | 1318 | 1483 |
| non alignements N | 39272 | 39099 | 38957 | 38896 | 38562 |
| collisions | 100 | 101 | 111 | 112 | 116 |
| VP | 1052 | 1133 | 1197 | 1224 | 1385 |
| FP | 83 | 88 | 90 | 93 | 97 |
| TN | 36860 | 36849 | 36840 | 36833 | 36823 |
| FN | 2142 | 1980 | 1852 | 1798 | 1474 |
| **precision** | **92.61%** | **92.72%** | **92.93%** | **92.87%** | **93.39%** |
| **rappel** | **32.94%** | **36.40%** | **39.26%** | **40.50%** | **48.44%** |

## Exact

On aligne les ouvrages de 2 sources si leur clé sont exactement les mêmes.

A screenshot of a computer

Description automatically generatedA blue and white rectangular object with a black text

Description automatically generated

## Approximatif

On aligne les ouvrages de 2 sources si leur clé suffisamment similaires (supérieures ou égales à un ratio de similarité), et on prend la clé qui a le ratio le plus élevé pour l’aligner (cf. <https://github.com/KarimLak/Books-Python/blob/96f7f4664697ff17286183fec63f8480e6ec7e53/andre/utils.py#L30> )

A screenshot of a computer

Description automatically generatedA screenshot of a quiz

Description automatically generatedA screen shot of a math equation

Description automatically generatedIci, notre ratio minimal est de 0.95, aucune des clés ne va être alignée (car le max est de 0.9).

Si notre ratio minimal est de 0.9, on va prendre la clé qui a le plus haut ratio de similarité à a\_b\_c parmi les clés de BNF. Ici on a 2 clés candidates qui ont le même ratio dans BNF, donc on en prend arbitrairement une (la 1ere qui a eu le ratio le plus élevé.

### Complexité

C’est un calcul de la plus longue sous-séquence similaire, une variation de Ratcliff-Obershelp.

Calcul de ratio : O(m\*n) avec m et n les longueurs des 2 clés à comparer

On effectue ce calcul de ratio pour toutes les combinaisons de clés entre les 2 exemplaires ce qui nous donne une complexité finale de O(x\*y\*m\*n) avec x et y les tailles des 2 exemplaires.

### Parallélisation

J’ai utilisé de la parallélisation car cet algorithme de calcul de ratio prend du temps (joblib, Parallel).

Je réutilise mon pool de thread pour ne pas la recréer à chaque itération :

<https://github.com/KarimLak/Books-Python/blob/96f7f4664697ff17286183fec63f8480e6ec7e53/andre/db_alignments_approx.py#L81>

Comme documenté ici :

<https://joblib.readthedocs.io/en/latest/parallel.html#reusing-a-pool-of-workers>

J’ai isolé la fonction à paralléliser pour pouvoir la passer à des workers qui n’utilise pas de mémoire partagée :

<https://github.com/KarimLak/Books-Python/blob/96f7f4664697ff17286183fec63f8480e6ec7e53/andre/interdbstats_approx.py#L23>

M et N sont de taille négligeable par rapport à X et Y, mais on passe tout de même de quelques minutes d’exécutions à >8h d’exécution en single thread (temps exact pas connu car pas essayé) et 5h30 d’exécution en multithread avec 6 cœurs et 12 threads.

Par livre, on passait de 0.001 secondes d’alignement exact à environ 1 seconde par livre en single thread et 0.2 secondes en multithread (qui se dégradait à 1 seconde à cause de la température).

### Améliorations

<https://stackoverflow.com/questions/6690739/high-performance-fuzzy-string-comparison-in-python-use-levenshtein-or-difflib>

On pourrait utiliser Levenshtein qui a l’air plus rapide que difflib, Jaccard ou minhashing (approximation Jaccard)

[probability - Minhashing on Strings with K-length - Stack Overflow](https://stackoverflow.com/questions/47577204/minhashing-on-strings-with-k-length)

On pourrait également utiliser un filtre de bloom pour voir si un ouvrage est présent dans la base de données (ça serait une optimisation pour réduire les accès disques si la base de données devient très utilisée) .

## Hybride

ISBN & exact pour Constellation & BNF

Raison : prend du temps à faire l’alignement approximatif pour ce jeu de donné et on a déjà l’ISBN

Exact & approximatif pour Lurelu

Raison : pas d’ISBN pour Lurelu

## Collisions

Pour tous les algorithmes, les clés qui rentrent en collision avec une clé déjà alignée sont « [oubliées](https://github.com/KarimLak/Books-Python/blob/558fb2a9fccbc238e8b1afed3298808b0ba393f9/andre/interdbstats_exact.py#L21)». La raison des collisions sont les doublons à l’intérieur de chaque base de données.

# Format fichiers alignements

## Colonnes CSV

J’ai choisi « { » comme délimiteur csv à cause des caractères spéciaux dans la clé des livres avant mon prétraitement. Ce n’est peut-être plus nécessaire, et peut-être changer dans la fonction output\_csv

|  |  |
| --- | --- |
| key | la clé de l'ouvrage |
| key used to align BNF | clé utilisée pour aligner BNF à Constellation |
| isbn\_constellation |  |
| isbn\_bnf |  |
| isbn\_lurelu |  |
| key used to align lurelu | clé utilisée pour aligner Lurelu à Constellation et/ou BNF |
| similarity ratio | ratio de similarité entre Lurelu et Constellation/BNF (car alignement approximatif que pour Lurelu ) |
| url\_constellation |  |
| url\_bnf |  |
| url\_lurelu |  |

## Champs RDF

|  |  |
| --- | --- |
| pbs:exact\_key | la clé de l'ouvrage |
| pbs:uri\_bnf | la référence rdf de l'ouvrage BNF (peut être vide) |
| pbs:uri\_lurelu | la référence rdf de l'ouvrage Lurelu (peut être vide) |
| pbs:uri\_constellation | la référence rdf de l'ouvrage Constellation (peut être vide) |
| ns1:author | information du livre non traitée |
| ns1:datePublished |
| ns1:isbn\_constellation |
| ns1:isbn\_bnf |
| ns1:isbn\_lurelu |
| ns1:name |

A terme, on pourrait aussi mettre les âges selon les différentes sources (ou une unification de ces âges). De même pour les prix, et les critiques. Cela permettra une recherche rapide à partir des alignements.

# Alignements BNF & Constellation

## Résultats

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **clé** | **n\_a** | **n\_a\_e** | **n\_a\_e\_d** | **n\_a\_d** | **n\_a\_e\_d** |
| **Alignement algo** | **Exact** | **Exact** | **Exact** | **Exact** | **Approx\_.9** |
| Alignement corrects (VP) | 1333 | 1052 | 1071 | 1389 | 1366 |
| Alignements incorrects (FP) | 691 | 410 | 47 | 97 | 519 |
| Non-alignements corrects (VN) | 34934 | 35860 | 36942 | 36821 | 34260 |
| Non-alignements incorrects (FN) | 1229 | 1922 | 2109 | 1466 | 1228 |
| **Précision** | **65,79 %** | **71,81 %** | **95,63 %** | **93,41 %** | **72,39 %** |
| **Rappel** | **52,03 %** | **35,37 %** | **33,68 %** | **48,65 %** | **52,66 %** |

## Conclusions

Précision bonne mais rappel à améliorer pour les clé précises.

# Alignements BNF, Constellation & Lurelu

## Résultats

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **clé** | **n\_a\_e\_d** | **n\_a\_e\_d** | **n\_a\_e\_d** | **n\_a\_e\_d** |
| Alignement algo Lurelu | Exact puis approx\_.95 | Exact puis approx\_.9 | Exact puis approx\_.8 | Approx\_.9 |
| Alignement corrects (VP) | 64 | 74 | 82 | 74 |
| Alignements incorrects (FP) | 6 | 8 | 14 | 8 |
| Non-alignements corrects (VN) | 38985 | 38982 | 38969 | 38979 |
| Non-alignements incorrects (FN) | 107 | 85 | 59 | 85 |
| **Précision** | **91,43 %** | **87,06 %** | **76,64 %** | **87,06 %** |
| **Rappel** | **37,43 %** | **46,54 %** | **58,16 %** | **46,54 %** |

## Analyse

L’algorithme hybride est une version plus rapide de l’approximative, car quand on a une correspondance exacte entre les clés, on ne va pas faire le calcul cher de l’algorithme approximatif car on obtiendrai le même alignement (car l’approximatif va trouver un ratio de 1 avec la clé exacte et la sélectionner).

C’est pour cela qu’on observe la même performance entre l’hybride avec ratio de 0.9 et l’approximatif 0.9.

## Erreur(s) ISBN Lurelu

Après avoir repassé dans les données, ai remarqué un alignement qui n’avait pas de sens :

|  |  |  |  |
| --- | --- | --- | --- |
| clé | isbn\_constellation | isbn\_bnf | isbn\_lurelu |
| frissonlecureuil\_melaniewatt\_scholastic\_2006 | 439940478 | N/A | 2747021076 |
| frissonlecureuil\_melaniewatt\_bayard\_jeunesse\_2006 |  | 2747021076 |  |

Mais pourquoi Lurelu n’est-il pas aligné avec BNF ? S’ils ont le même ISBN, ils sont censés avoir la même clé ?

En fait, la clé de Lurelu est « frissonlecureuil\_melaniewatt\_scholastic\_2006 », donc on a un faux-positif à cause d’une erreur d’annotation des données : l’ISBN de Lurelu aurait dû être 439940478.

Nous devrions faire une vérification des annotations Lurelu.

https://constellations.education.gouv.qc.ca/index.php?p=il&lo=34179

https://catalogue.bnf.fr/ark:/12148/cb402369231

<https://www.lurelu.net/coupsdecoeur03.html>

## A-t-on besoin de l’alignement hybride si on a les ISBN ?

Parfois, les livres semblent être les même dans Constellation et BNF, mais ont des ISBN et EAN différents :

https://catalogue.bnf.fr/ark:/12148/cb42626519f

https://constellations.education.gouv.qc.ca/index.php?p=il&lo=42044

Ces livres sont alignés avec la clé exacte n\_a\_e\_d, mais l’alignement est considéré comme un faux positif à cause de la différence d’ISBN.

Il serait intéressant de demander aux utilisateurs s’ils veulent que ces livres soient alignés ou non.

# Organisation des fichiers

Dans le repo de Karim, sous le dossier « [Andre](https://github.com/KarimLak/Books-Python/blob/96f7f4664697ff17286183fec63f8480e6ec7e53/andre/utils.py)»:

## Code source

|  |  |  |  |
| --- | --- | --- | --- |
| nom | responsabilités | input | output |
| *stats\_données\_intra.py* | - générer données manquantes et doublons de clés à partir des données rdf  - calcul de similarité entre les âges des doublons | *output\_bnf.ttl*  *output\_constellations.ttl* | *BNF\_intra\_stats.csv* = lien des données manquantes dans BNF  *BNF\_doublons\_<type\_de\_clé>\_intra\_stats.csv* = doublons à l’intérieur de BNF *<type\_de\_clé>* comme clé  *Constellation\_intra\_stats.csv* = lien des données manquantes dans BNF  *Constellation\_doublons\_<type\_de\_clé>\_intra\_stats.csv* = doublons à l’intérieur de BNF *<type\_de\_clé>* comme clé |
| *utils.py* | - fonctions pour extraire les données RDF de chaque source  - création de la clé  - prétraitement de la clé |  |  |
| *db\_alignments.py* | alignement exact grossier avec plusieurs types de clés entre BNF & Constellation | *output\_bnf.ttl*  *output\_constellations.ttl* | *stats.log* : stats et matrice de confusion  *<type\_de\_clé>.csv*  -> alignement avec le type de clé |
| *db\_alignments\_approx.py* | alignement approximatif entre BNF & Constellation  !! consomme 100% du cpu pendant une longue durée!! | *output\_bnf.ttl*  *output\_constellations.ttl* | *approx\_alignment\_<type\_de\_clé>\_ratio\_<ratio\_similarité>*  *approx\_<key\_name>\_ratio\_<ratio\_similarité >\_execution\_time.log*  -> logfile pour le temps d’exécution de chaque livre  *approx.\_<key\_name>\_ratio\_<ratio\_similarité >\_stats.log*  -> logfile pour les statistiques et la matrice de confusion |
| *db\_alignments\_hybrid.py* | alignement ISBN puis exact entre BNF & Constellation, ainsi qu’exact puis approximatif entre Lurelu et Constellation/BNF | *output\_bnf.ttl*  *output\_constellations.ttl*  *output\_lurelu.ttl* | *hybrid\_alignment\_<type\_de\_clé> \_ratio\_\_< ratio\_similarité >.csv*  *hybrid\_alignment\_{self.key\_type}\_ratio\_<ratio\_similarité>.ttl*  -> données rdf  *hybrid\_<type\_de\_clé>\_ratio \_<ratio\_similarité>\_stats.log*  *hybrid\_<type\_de\_clé>\_ratio\_<ratio\_similarité>\_execution\_time.log* |
| *interdbstats.py*  *interdbstats\_exact.py*  *interdbstats\_approx.py*  *interdbstats\_hybrid.py* | classes pour aligner et faire les statistiques  -> diagramme de classe dans *class\_diagram.drawio* |  |  |
| *bookalignment*.py | représente 1 alignement |  |  |
| *alignement\_rdf\_parsing.py* | code minimal pour lire le fichier d’alignements rdf | *hybrid\_alignment\_{self.key\_type}*  *\_ratio\_<ratio\_similarité>.ttl* |  |
| *fusion\_ages\_algos\_exploration.py* | quelques idées basiques pour uniformiser les âges |  |  |

## Données

Dans le repo de Karim, sous le dossier « Andre/data»:

<https://github.com/KarimLak/Books-Python/blob/96f7f4664697ff17286183fec63f8480e6ec7e53/andre/data>

|  |  |
| --- | --- |
| nom | contenu |
| light stats | - database légère pour tester les algos - output des algos |
| data as of end of july | - output des algos avec les - données de fin juillet |
| data as of 4 august | - données du 4 aout pour reproduire les résultats  - essais des différentes méthodes de prétraitement (avec alignement exact)  - alignement approx n\_a et n\_a\_d (car éditeurs pas terminés le 4 aout) |
| data as of 8 august | - données du 8 aout sont les données finales (situées [ici](https://github.com/KarimLak/Books-Python/blob/8b65d002631f3e1ccecf0dc84bd1045b4e96a3ee/final_datasets) )  - alignement hybrides et approx avec constellation, BNF et lurelu |