1-2-Constituer_un_corpus

Tout travail de recherche impliquant du texte nécessite en premier lieu de constituer un corpus d'étude, c'est à dire de réunir un ensemble de documents (articles, extraits, transcription...) autour d'une thématique commune.
La constitution d'un corpus reste une tâche longue, d'autant plus qu'elle implique souvent des étapes supplémentaires pour préparer le corpus à l'analyse.

Dans ce cours, nous allons apprendre à constituer un corpus à partir de documents sur le Web, et à appliquer des traitements à ces données pour préparer le corpus à de futures analyses. Pour cela, nous allons employer l'outil open source et gratuit OpenRefine (https://openrefine.org).

Installer OpenRefine

Suivez les instructions correspondant à votre système d'exploitation (Windows, Mac, Linux) indiquées sur le site

Ouvrez OpenRefine une fois le logiciel installé. Un onglet s'ouvrira dans votre navigateur. La page d'accueil permet de créer de nouveaux projets.

Créer un projet

https://openrefine.org/docs/manual/starting

OpenRefine permet de créer un projet à partir de données existantes. Il existe plusieurs façons d'importer les données :

- chargeant un fichier depuis l'ordinateur
- en fournissant des liens URLs
- en copiant-collant les données (Clipboard)
- depuis une base de données SQL
- depuis Google Sheets ou Google Drive

OpenRefine accepte des fichiers dans différents formats :

- comma-separated values (CSV) ou text-separated values (TSV)
- fichiers textes (.txt)

- Fixed-width columns
- JSON
- XML
- OpenDocument spreadsheet (ODS)
- Excel spreadsheet (XLS or XLSX)
- PC-Axis (PX)
- MARC
- RDF data (JSON-LD, N3, N-Triples, Turtle, RDF/XML)
- Wikitext

Un aperçu des données s'ouvre après les avoir importées. Un onglet en bas de l'interface permet de choisir le format, afin d'aider OpenRefine a traiter les données. Il est recommandé de fournir des données au format .csv ou .tsv, l'interface d'OpenRefine étant également présentée sous forme de tableau.

Un boîte de texte en haut à droite permet d'indiquer le nom du projet, ainsi que d'importer les données.

Cliquez sur `This Computer > Select files`, sélectionnez le fichier `corpus.csv` puis cliquez sur `Next`

Vous accédez à une page qui vous donne un aperçu de votre projet. Vous devriez voir un tableau avec la même structure et le même contenu que le fichier `corpus.csv`.

La partie basse de la page permet de choisir différents formats pour traiter les données. Puisque ici notre fichier est au format .csv, il nous faut choisir l'option `CSV / TSV / separator-based files`. Cependant, d'autres formats sont possibles. Cliquez sur le bouton `Create project` pour continuer.

Explorer les données

https://openrefine.org/docs/manual/exploring

Types de données

Les données dans les colonnes de OpenRefine peuvent correspondre à plusieurs types :

- **text** (pour les valeurs textuelles)
- **numbers** (pour les valeurs numériques)
- **bool** (pour les valeurs booléennes, cad. True ou False)

- **date** (pour les valeurs temporelles)
- **error** (s'il y a eu une erreur dans le traitement de la cellule)
- **null** (si la cellule est vide)

Les dates doivent respecter un format particulier, qui est le format ISO-8601. Ainsi, elles doivent suivre la structure YYYY-MM-DDTHH:MM:SSZ. OpenRefine permet de convertir les cellules d'une colonne dans ce format, il n'y a pas besoin de le faire manuellement. Les dates en informatiques nécessitent de respecter des formats bien précis. N'hésitez donc pas à consulter la documentation de OpenRefine pour maîtriser ce type de données.

Pour convertir les valeurs de la colonne `date`, cliquez sur la flèche de la colonne `date` puis sur `Edit cells > Common transforms > To date`

Lignes et entrées (rows et records)

Par défaut, les données sont représentées en **lignes** (row), comme dans un tableau classique. Cependant, plusieurs lignes peuvent appartenir au même groupe (par exemple, avoir plusieurs correspondant aux tags HTML du contenu d'une page Web). Dans ce cas, on peut observer les données par **entrée** (records). Les deux options `row` et `record` se situent en haut à gauche de l'interface.

Sort and view

https://openrefine.org/docs/manual/sortview

Chaque colonne possède une option `Sort` qui permet de trier les lignes en fonctions de certains paramètres. Il est possible de trier les valeurs d'une colonne selon les types de données, et en choisissant par ordre croissant comme décroissant. On peut réorganiser les blocs sur la boîte de droite pour choisir dans quel ordre apparaissent les valeurs complètes, les erreurs et les cellules vides.

Cliquez sur la flèche de la colonne `length` puis sur `Sort > Sort...`. Choisissez l'option `numbers` et `largest first`, puis cliquez sur `Ok`. Vous devriez voir l'ordre des données changer.

Un bouton `Sort` apparait en haut de l'interface, dans lequel les différents paramètres de tris apparaissent. Ainsi, il est possible de trier les données selon plusieurs paramètres.

Essayez d'appliquer des tris à différentes colonnes du tableau, et observez les effets. Pour supprimer un tri, cliquez sur `Sort` en haut de l'interface, choisissez la ligne avec le nom de la colonne qui vous intéresse, puis sur `Remove sort`.

Les colonnes possèdent une option `View`, qui permet de filtrer les colonnes visibles. Les options avec `Collapse` permettent de faire disparaître des colonnes, tandis que les options avec `Expand` permettent de les faire apparaître.

Choisissez une colonne, cliquez sur sa flèche, puis sur `View`. Essayez les différentes options proposées, et observez les effets.

Annuler et Refaire des étapes

Vous pouvez à tout moment annuler une action que vous avez réalisée sur le corpus. Pour cela, cliquez sur `Undo / Redo` en haut à gauche de l'interface. Vous verrez apparaître dessous la liste des actions réalisée. Cliquez sur l'action que vous souhaitez pour revenir à cette étape.

Facets

https://openrefine.org/docs/manual/facets

Les facettes (facets) permettent d'observer les données selon une perspective précise. Plus concrètement, il s'agit d'observer les données par rapport à une colonne en particulier, par exemple en observant toutes les lignes ayant une valeur commune...

Facette textuelle (Text facet)

Les facettes textuelles permettent d'appliquer une facette aux colonnes contenant des données textuelles. Elle apparaît sous forme de tableau à gauche de l'interface, dans lequel les occurrences de chaque élément textuel de la colonne apparaissent.

Cliquez sur la flèche de la colonne `content` puis sur `Facet > Text Facet`. La facette apparaîtra à gauche de l'interface.

On peut trier par ordre alphabétique ou par fréquence en changeant l'option `Sort`.

Cette facette est très utile pour étudier les fréquences des catégories, mais aussi utiles pour identifier les doublons.

Un bouton `edit` apparaît en survolant chaque élément de la facette avec la souris. Ce bouton permet de modifier la valeur de toutes les cellules qui ont cette valeur (idéal par exemple pour corriger des fautes d'orthographes, ou appliquer des corrections à plusieurs éléments du corpus).

Cliquez sur le bouton `edit` du premier élément de la facette textuelle, puis ajouter le mot `ajout` au texte, et cliquez sur valider.

Les éléments de la facette ont également un bouton `exclude` qui permet d'exclure les éléments correspondants du corpus.

Cliquez sur le bouton `exclude` du premier élément de la facette textuelle pour voir les effets.

Facette numérique (Numeric facet)

Les facettes numériques permettent d'appliquer une facette aux colonnes contenant des données numériques. Cette facette apparaît sous la forme d'un histogramme sur la gauche de l'interface. Cet histogramme possède des curseurs qui permettent de sélectionner des portions de données.

Cliquez sur la flèche de la colonne `length` puis sur `Facet > Numeric Facet`. La facette apparaîtra à gauche de l'interface.

Facette temporelle (Timeline facet)

Les facettes temporelles permettent d'appliquer une facette aux colonnes contenant des dates. Comme pour les facettes numériques, elles apparaissent sous forme d'histogramme sur la gauche, où des curseurs permettent de filtrer les données en fonction des dates.

Cliquez sur la flèche de la colonne `date` puis sur `Facet > Date Facet`. La facette apparaîtra à gauche de l'interface

Facette personnalisée

Les facettes personnalisées permettent d'appliquer des facettes plus précises, notamment à l'aide d'expressions GREL (voir plus bas).

Cliquez sur la flèche de la colonne `content` puis sur `Facet > Custom text facet`. Dans le champ `Expression`, écrit `value.contains("recevez")`. Observez les résultats obtenus.

Transformer les données

Les données contenues dans le tableau ne sont pas immuables. Au contraire, OpenRefine met à disposition de nombreux outils pour modifier les données au niveau des cellules, des lignes et

des colonnes.

Edition de cellules

https://openrefine.org/docs/manual/celleditingPour éditer les cellules d'une colonne, cliquez sur la flèche d'une colonne puis sur `Edit cells` .

L'option `Common Transform` propose une liste de transformations fréquemment employées. Par exemple :

- convertir les valeurs d'une colonne dans un autre type de donnée
- enlever les espaces de trop
- échapper les caractères HTML
- changer la casse du texte

L'option `Transform` permet de modifier le contenu des cellules à l'aide d'expression GREL (voir plus bas), et donc d'appliquer des transformations qui ne sont pas proposées dans `Common Transform` .

Les options `Split multi-valued cells` et `Join multi-valued cells` permettent respectivement de séparer et de joindre le contenu des cellules par rapport à un caractère (par exemple, séparer le contenu textuel au niveau des espaces, joindre les éléments d'une cellule par des virgules)

L'option `Cluster and edit` permet de grouper les données similaires.

L'option `Replace` permet de chercher un élément dans les cellules et de le remplacer par un autre élément (similaire à Contrôle-F).

Edition de colonnes

https://openrefine.org/docs/manual/columnediting

Comme il est possible d'éditer le contenu des cellules, il est également possible d'éditer les colonnes du corpus. Pour cela, cliquez sur la flèche de la colonne choisie puis sur `Edit column` . Les opérations possibles consistent principalement à fusionner des colonnes ensemble, ou bien au contraire à produire de nouvelles colonnes en fonction du contenu de la colonne sélectionnée.

L'option `Split into several columns` permet de séparer le contenu des cellules en fonction d'un caractère, et de créer des colonnes pour chaque élément. A l'inverse, l'option `Join columns...` permet de fusionner plusieurs colonnes.

L'option `Add column based on this column` permet de créer de nouvelles colonnes par rapport au contenu d'une colonne sélectionnée. Cette option nécessite l'emploi d'expressions GREL (voir plus bas).

L'option `Add column by fetching URLs` permet de collecter le contenu de pages web à partir d'une colonne contenant des liens URLS.

Enfin, il existe d'autres options pour renommer et déplacer les colonnes.

Expressions

https://openrefine.org/docs/manual/expressions

Les expressions correspondent à du code permettant de manipuler les données au delà des options de bases proposées par OpenRefine. De nombreux traitements des données sont possibles grâce aux expressions. Les expressions sont principalement exprimées par le langage **GREL (General Refine Expression Language)**, propre à OpenRefine, mais d'autres langages de programmation peuvent être employés, comme Python ou Clojure.

Les expressions sont utilisables pour les opérations suivantes :

- `Facet`
- `Edit cells`
- `Edit columns`

Toutes ces opérations ouvrent une fenêtre, dans laquelle on trouve entre autre une champ `Expression`.

Au fur et à mesure que vous écrirez cette expression, vous verrez un aperçu du résultat dans le tableau dessous. Une fenêtre apparaitra également s'il y a une erreur dans l'expression.

Les expressions GREL

Les expressions commencent par une variable, qui peut avoir une des valeurs suivantes :

`value`	**La valeur de la cellule dans la colonne actuelle de la ligne actuelle (peut être nulle)**
`row`	La ligne actuelle

`value`	**La valeur de la cellule dans la colonne actuelle de la ligne actuelle (peut être nulle)**
`row.record`	Une ou plusieurs lignes regroupées pour former une entrée
`cells`	Les cellules de la ligne actuelle, avec les champs correspondant aux noms des colonnes (row.cells)
`cell`	Cellule de la colonne actuelle de la ligne actuelle, contenant la valeur de cellule et d'autres attributs
`cell.recon`	Informations de rapprochement de la cellule renvoyées par un service ou un fournisseur de services de rapprochement
`rowIndex`	La valeur de l'index de la ligne actuelle (la première ligne est 0)
`columnName`	Le nom de la colonne de la cellule courante, sous forme de chaîne de caractères

Les onglets `History`, `Starred` et `Help` permettent respectivement d'accéder à l'historique des expressions employées précédemment, d'accéder aux expressions favorites, et d'accéder à la documentation de GREL.

Fonctions

Les fonctions sont des éléments de code permettant de réaliser des actions pré-définies. Elles sont reconnaissables au fait qu'elles se terminent par des parenthèses, qui peuvent contenir des arguments supplémentaires. Les fonctions utilisables dépendent des types de données. Par exemple, les fonctions des données textuelles ne peuvent être employées par les données numérique.

Des exemples de fonctions par types de données :

- Chaîne de caractères (String)
  - `length()` retourne la taille du texte en nombre de caractères
  - `contains(a)` vérifie que le texte continue l'élément a
  - `toLowercase()` transforme le texte en minuscule
  - `replace(a, b)` remplace le texte a par le texte b
- Liste (Array)
  - `length()` taille de la liste en nombre d'éléments
  - `join(a)` assemble les éléments de la liste par le caractère a
  - `uniques()` identifie tous les éléments d'une liste
- Date
  - `toDate()` transforme la valeur au format date
- Math

    - `cos(a)` retourne le cosinus de a
    -
- Basée sur le format
    - `parseHtml()` permet de lire la structure d'un format HTML
    - `select(a)` sélectionne une balise dans la structure HTML
    - `wholeText()` retourne le contenu textuel d'une balise HTML

Pour appliquer une fonction à une valeur, on emploie une notation à base de points (dot notation). Par exemple, pour appliquer la fonction `length()`, on écrira `value.length()`. Il est possible (même nécessaire parfois) d'enchaîner l'application de fonctions. Par exemple, pour obtenir la longueur d'une liste obtenue à partir d'un texte que l'on aura divisé au niveau des espaces, on écrira `value.split(" ").length()`.

Cas pratique : collecter le contenu des pages webs

Cliquez sur la flèche à côté de "url" dans le tableau puis `Edit column > Edit column by fetching URLs`. Une nouvelle fenêtre s'ouvre. Dans le champ `New column name`, entrez la valeur `fetch`, puis cliquez sur `Ok`. Une nouvelle colonne doit apparaître, dans laquelle est contenue la page HTML correspondant à l'URL donné.

Ces pages HTML contiennent de nombreux tags. Cependant, le contenu des articles est situé entre des balises `<p>`. Cliquez sur la flèche vers la colonne `fetch`, puis `Edit column > Add column based on this column`.Dans le champ `New column name`, entrez la valeur `content`. Dans la champ `Expression`, entrez la valeur suivante : `value.parseHtml().select("p").join("|")`.

Explication de l'expression :

- `value` correspond à la valeur de la cellule
- `parseHtml()` est une fonction permettant de lire un document HTML. Si besoin, la fonction transforme le contenu de la cellule en HTML pour pouvoir la lire
- `select()` est une fonction permettant d'indiquer quel élément on souhaite sélectionner. Ici on sélectionne les balises `p`. Notez qu'il faut indiquer la balise entre guillemets `"`. Cette fonction retourne une liste de tous les éléments identifiés
- Une cellule ne peut contenir de liste d'éléments. Elle ne peut contenir que des éléments selon les quatre types possibles. Pour transformer la liste en texte, on emploie la fonction `join()`. On indique entre guillemets que les éléments de la liste sont joints par le symbole `|`.

Cliquez sur `Ok` une fois terminé.

Cliquez sur la flèche de la colonne `content`, puis sur `Edit cells > Split multi-valued cells`. Choisissez l'option `by separator`, et indiquez le caractère `|` dans le champ `Separator`, puis cliquez sur `Ok`. Vous pouvez voir alors que tous les éléments `<p>` sont séparés sur des lignes différentes.

Cliquez sur la flèche de la colonne `content`, puis sur `Edit cells > Transform`. Dans le champ `Expression`, entrez : `value.parseHtml().wholeText()`

- `parseHtml()` transforme la valeur de la cellule en HTML
- `wholeText()` permet de récupérer le contenu textuel, en ignorant les balises HTML
  Cliquez sur `Ok`. Le contenu des cellules ne doit alors plus que être textuel, sans balise HTML.

Cliquez sur la flèche de la colonne `content`, puis sur `Edit cells > Common transforms > Unescape HTML entities`. Cela permet de supprimer certains caractères spéciaux propres au HTML.

Cliquez sur la flèche de la colonne `content`, puis sur `Edit column > Add column based on column content`. Indiquez `length` dans le champ `New column name` puis dans le champ `Expression`, indiquez `value.length()`

- La fonction `length()` vous donne la taille en nombre de caractère du contenu de la cellule

Cliquez sur la flèche de la colonne `content`, puis sur `Edit column > Add column based on column content`. Indiquez `Recevez` dans le champ `New column name` puis dans le champ `Expression`, indiquez `value.contains("Recevez")`.

- La fonction `contains()` vous indique si un élément textuel est présent dans le contenu de la cellule. Il faut indiquer cet élément entre guillemets
  Les valeurs du champ `recevez` sont soit `true` soit `false`. On parle ici de valeur `booléenne`. Les valeurs du champ `length` sont de type numérique, tandis que les valeurs du champ `content` sont de type texte.

Cliquez sur la flèche de la colonne `content`, puis sur `Edit column > Add column based on column content`. Indiquez `unique` dans le champ `New column name` puis dans le champ `Expression`, indiquez `value.split(" ").uniques().join(",")`.

- `split()` permet de diviser une texte au niveau d'un caractère en particulier (ici les espaces)
- `uniques()` retourne les éléments uniques d'une liste (ici, une liste de mots obtenus par la fonction `split()`)
- On emploie la fonction `join()` pour obtenir un texte à partir de la liste, cette fois en séparant les éléments par une virgule.

Cliquez sur la flèche de la colonne `date`, puis sur `Edit cell > Transform`. Dans le champ `Expression`, entrez `value.toDate()` puis cliquez sur `Ok`.

- La fonction `toDate()` permet de convertir le contenu d'une cellule du format texte au format date.

Cliquez sur la flèche de la colonne `unique` puis sur `Edit column > Split into several columns`. Dans le champ `Separator`, indiquez `,` (virgule) comme séparateur puis cliquez sur `Ok`. Observez les résultats. N'hésitez pas à cliquer sur `Undo` pour annuler cette opération si besoin.

Vous pouvez sélectionner des colonnes que vous souhaitez conserver. Pour cela, cliquez sur la flèche de la colonne `All` (la plus à gauche de l'interface), puis `Edit columns > Re-order / remove columns...`. Dans la fenêtre qui s'ouvre, déplacer les noms de colonnes dans la boîte de droite pour les supprimer, et laisser les noms dans la boîte de gauche pour les conserver. Par exemple, déplacez tous les noms de colonnes dans la boîte de droite sauf la colonne "content", puis faites `Ok`.

Exporter les données

https://openrefine.org/docs/manual/exporting

OpenRefine permet d'exporter le corpus sous différents, afin de pouvoir le réemployer pour d'autres tâches ou avec d'autres logiciels. Attention, il exportera les valeurs visibles du tableau, c'est à dire que les facets, tris,... sont pris en compte pour l'export des données.

Cliquez sur `Export` en haut à droite, puis sur `comma-separated value` pour exporter votre travail au format .csv. Attention, il semblerait que OpenRefine structure les données au format .csv, mais les enregistre au format .txt. Pensez donc à renommer votre fichier, et à remplacer .txt par .csv.

Pour aller plus loin

- Ensemble de recettes d'expressions GREL : https://github.com/OpenRefine/OpenRefine/wiki/Recipes
- Réconciliation, association à bases de données externes
- Ajouter des colonnes en fonction de valeurs réconciliées
- Installer des extensions pour importer d'autres formats
- Apprendre les structures de contrôles (if, else), les boucles (for), les opérateurs logiques
- Explorer les fonctions disponibles pour les expressions

- Employer des expressions avec des variables autres que `value`
- Employer des expressions régulières dans les Expressions
- Clusters et éditions
- Traiter les cellules vides
- Transposer les cellules : https://openrefine.org/docs/manual/transposing

Ressources

Programming Historian

- https://programminghistorian.org/en/lessons/fetch-and-parse-data-with-openrefine