

1 Modèle basé sur les données manuellement étiquetées

L'objet de ce notebook est de mettre en place le modèle basé sur les données manuellement étiquetées.

1.1 Récupération des données

1.1.1 Préambule technique

```
[1]: # setting up sys.path for relative imports
from pathlib import Path
import sys
project_root = str(Path(sys.path[0]).parents[1].absolute())
if project_root not in sys.path:
    sys.path.append(project_root)

[2]: # imports and customization of display
import os
import pandas as pd
pd.options.display.min_rows = 6
pd.options.display.width = 108
pd.options.display.latex.repr = True
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline

from src.pimapi import Requester
from src.pimest import ContentGetter
from src.pimest import PathGetter
from src.pimest import PDFContentParser
from src.pimest import BlockSplitter
from src.pimest import SimilaritySelector

[3]: # monkeypatch _repr_latex_ for better inclusion of dataframes output in report
def _repr_latex_(self, size='scriptsize',):
    return(f"\\resizebox{{{\\linewidth}}}{!}}{{\\begin{{{size}}}}\\centering{{{self.
↳to_latex()}}}{\\end{{{size}}}}}")
pd.DataFrame._repr_latex_ = _repr_latex_
```

1.1.2 Chargement du fichier des données manuellement étiquetées

On commence par charger le fichier csv contenant les données manuellement étiquetées.

```
[4]: ground_truth_df = pd.read_csv(Path('.') / '..' / 'ground_truth' / 'manually_labelled_ground_truth.csv',
                                   sep=';',
                                   encoding='latin-1',
                                   index_col='uid')

ground_truth_df.head()
```

```
[4]:
```

uid	designation	ingredients
a0492df6-9c76-4303-8813-65ec5ccbfa70	Concentré liquide Asian en bouteille 980 ml CHEF	Eau, maltodextrine, sel, arômes, sucre, arôme ...
d183e914-db2f-4e2f-863a-a3b2d054c0b8	Pain burger curry 80 g CREATIV BURGER	Farine de blé T65, eau, levure, vinaigre de ci...
ab48a1ed-7a3d-4686-bb6d-ab4f367cada8	Macaroni en sachet 500 g PANZANI	- 100% Semoule de BLE dur de qualité supérieur...
528d4be3-425c-4f8b-8a87-12f1bc645ddd	Fève de Tonka en sachet 100 g COMPTOIR COLONIAL	fève de tonka (graines ridées de 25 à 50mm de ...
51b38427-b2ea-4c56-93e8-4242361ef31b	Caviar d'aubergine en pot 500 g PUGET RESTAURA...	Aubergine 60,5% (aubergine, huile de tournesol...

```
[5]: ground_truth_uids = list(ground_truth_df.index)
ground_truth_uids[:5]
```

```
[5]: ['a0492df6-9c76-4303-8813-65ec5ccbfa70',
      'd183e914-db2f-4e2f-863a-a3b2d054c0b8',
      'ab48a1ed-7a3d-4686-bb6d-ab4f367cada8',
      '528d4be3-425c-4f8b-8a87-12f1bc645ddd',
      '51b38427-b2ea-4c56-93e8-4242361ef31b']
```

1.1.3 Pipeline d'acquisition du contenu des données

On commence par construire un premier pipeline d'acquisition des données. Il fonctionne en 3 étapes : -
détermination du chemin vers lequel aller chercher les fiches techniques - récupération du contenu binaire du
fichier - conversion de ce contenu binaire en texte

```
[6]: acqui_pipe = Pipeline([('PathGetter', PathGetter(ground_truth_uids=ground_truth_uids,
                                                         train_set_path=Path('.') / '..' / 'ground_truth',
                                                         ground_truth_path=Path('.') / '..' / 'ground_truth',
                                                         )),
                           ('ContentGetter', ContentGetter(missing_file='to_nan')),
                           ('ContentParser', PDFContentParser(none_content='to_empty'))],
    verbose=True)
```

```
[7]: texts_df = acqui_pipe.fit_transform(ground_truth_df)
texts_df.sample(5)
```

```
[Pipeline] ... (step 1 of 3) Processing PathGetter, total= 0.1s
[Pipeline] ... (step 2 of 3) Processing ContentGetter, total= 0.6s
Launching 8 processes.
[Pipeline] ... (step 3 of 3) Processing ContentParser, total= 37.2s
```

[illegible]

On peut afficher quelques textes récupérés par le pipeline :

```
[8]: with pd.option_context("max_colwidth", 1000):
      pass
      #     print(texts_df.sample(3, random_state=42)['text'])
      #     (texts_df.sample(3, random_state=42)['text']
      #       .to_latex(Path('.') / 'tbls' / 'processed_FT.tex',
      #                 index=False,
      #                 index_names=False,
      #                 column_format='p{\linewidth}',
      #                 na_rep='-',
      #                 escape=True,
      #               )
      #     )
```

1.2 Découpage en blocs

On découpe les longs textes en blocs. Chaque texte devient une liste de strings plus court.

```
[9]: def splitter(text):  
      return(text.split('\n\n'))
```

```
[10]: split_transfo = BlockSplitter(splitter_func=splitter)
      splitted_df = split_transfo.fit_transform(texts_df)
      splitted_df.sample(5)
```

Launching 8 processes.

[illegible]

On peut afficher un exemple de texte découpé en blocs :

```
[11]: sep = '\n-----\n'
sample = splitted_df.sample(1, random_state=39)['blocks'].iloc[0]
print(sep.join(sample))

tex_str = (
    pd.DataFrame(sample, columns=['Blocs'])
```

```

        .to_latex(column_format='p{10cm}',
                    index=False,
                    index_names=False,
                    escape=True,
                    )
        .replace(r'\textbackslash n', '\\newline ')
    )

#with open(Path('.') / 'tbls' / 'block_example.tex', mode='w') as file:
#    file.write(sep.join(sample).replace('\n', r' \newline '))

```

30/12/19

Date d'impression :

Remarque :

Les informations contenues dans cette fiche technique sont données de bonne foi, en l'état actuel de nos connaissances, et selon les indications communiquées par le producteur ou le fournisseur. Il appartient au client de vérifier la conformité de la marchandise par rapport à l'usage qu'il en fait.

Création :

12/06/12

12 rue René Cassin
37390 NOTRE DAME

Tél :

02 47 85 55 00

Fax :02 47 41 33 32

FICHE TECHNIQUE

Mélange du trappeur, 70 g

Trapper blend, 70g

Code article KEREX

Nom latin (si disponible)

/ EAN Code

Code barre

/ KEREX Code

/ (Latin name)

TEEPTRAPPEUR

X

3760063322262

Poids net

Poids brut

Origine

/ net weight

/ gross weight

/ Origin

0,07 Kilogramme

0,125 Kilogramme

CANADA

/ General information

Informations générales

DLUO conseillée / "Best before date" recommended

Nomenclature douanière / Customs code

Conditions idéales de stockage

/ Conditions of storage

Ingrédients :

Conserver dans un endroit frais et sec

Store in a cool dry place

5 ans / 5 years

0910999900

Sucre, poivre noir, coriandre, légumes déshydratés (ail, oignon, poivron rouge), sel de mer, sucre d'érable, arôme d'érable naturel, huile végétale (canola)

Sugar, black pepper, coriander, dehydrated vegetables (garlic, onion, red bell pepper), sea salt, maple sugar, natural maple aroma, vegetable oil (canola)

/ Ingredients

Contaminants / Contaminating

Ionisation / Irradiation

OGM / GMO

Pesticides/ Pesticides

Métaux Lourds

/ Heavy Metals

Allergènes et leurs dérivés (si présents)

/ Allergens (if existing)

Conformité à la directive 1999/2/CE (22/02/99)

Produit non ionisé et ne contenant pas d'ingrédients ionisés.

Not irradiated

accordingly with the Reg 1999/2/CE (22/02/99).

Free from GMO

Ne contient pas d'OGM, est non soumis à l'étiquetage sur les OGM

Conforme à la directive 396/2005 /CE

In accordance with Reg 396/2005 /CE.

Conforme au règlement 1881/2006 /CE

In accordance with Reg 1881/2006 /CE..

Gluten

Crustacés

Oeufs

Poisson

Soja

Lait

Fruits à coque - Arachides

Céleri

Moutarde

Sésame

Sulfites

Lupin

Mollusques

/ Gluten

/ Crustaceans

/ Eggs

/ Fish

/ Soy

/ Milk

/ Peanuts and Treenuts

/ Celery and celeriac

/ Mustarde

/ Sésame

```

/ Sulphites
/ Lupin
/ Shellfish
-----
Absence
Absence
Absence
Absence
Absence
Absence
Absence
Absence
Absence
Absence
Absence
Absence
Absence
-----
Caractères microbiologiques
-----
/ Microbiological characteristics
-----
Microorganismes aérobies 30 °C
Escherichia coli
Salmonelles
Levures
Moisissures
Aflatoxine Total
Aflatoxine B1
-----
/ Total plat count (APC)
E. Coli
/
/ Salmonella
/ Yeasts
/ Moulds
/ Total aflatoxin
B1 aflatoxin
/
-----
NF V05-051 < 6 000 000 / g
NF V08-053 < 10 / g
NF V08-052 Absence dans 25g
NF V08-059 < 10 000 / g
NF V08-059 < 10 000 / g
Kit Enzymatique < 10 ppb
Kit Enzymatique < 5 ppb
-----

```

1.3 Train / Test split

On procède au découpage en un jeu d'entraînement et un jeu de test en gardant 400 produits pour l'entraînement et 100 produits pour le test :

```
[12]: train, test = train_test_split(splitted_df, train_size=400, random_state=42)
```

1.4 Entraînement sur le jeu d'entraînement

On entraîne un modèle SimilaritySelector, sur le set d'entraînement :

```
[13]: model = SimilaritySelector()
```

```
[14]: model.fit(train['blocks'], train['ingredients'])
```

```
[14]: <src.pimest.SimilaritySelector at 0x7f048240de20>
```

```
[15]: len(model.count_vect.vocabulary_)
```

```
[15]: 1204
```

```
[16]: predicted = pd.Series(model.predict(test['blocks']),
                           index=test.index,
                           name='predicted'
                           )
predicted = pd.concat([test['ingredients'], predicted], axis=1)
predicted.sample(5)
```

```
[16]:
```

uid	ingredients	predicted
eadb972c-6623-472d-a11d-489a7faf6f11	- Soja fermenté naturellement (soja, sel, eau)...	Céréales contenant du gluten \net des produits...
6267b9f8-2529-4bc6-ba4b-26760f0522b3	eau gazéifiée\ncolorant : E150d\nacidifiants : ...	CocaCola Light mini 8 x 150 mlEAN544900023980...
04235024-80f3-46c2-bad0-aae0d5fab024	Persil	Céréales contenant du gluten (à savoir blé, se...
e51b7fd6-d878-47f8-a36b-f10f8d4087bd	Débris de truffes d'hiver, jus de truffes, sel	\nOrigine Truffes et Sel : ...
6566e18d-3bdd-43d8-ab0c-de51894621f9	Pommes 89%, purée de fraises à base de concentré...	Liste ingrédients : Pommes 89%, purée de frais...

```
[17]: predicted['pred_len'] = predicted['predicted'].apply(len)
sub_sample = predicted.loc[predicted['pred_len'] <= 500, ['ingredients', 'predicted']]
sub_sample.head(5)
```

```
[17]:
```

uid	ingredients	predicted
2892dd68-e3a6-474c-b543-3ebfd3490658	Café instantané, café torréfié moulu (3%).	- NESTLÉ a un système de management de la qual...
3634fb1e-ee79-41d1-8aaa-084c1fae5bd5	Poire 99,9%, antioxydant: acide ascorbique.	Ce produit est une purée de fruits obtenue à p...
345591f4-d887-4ddc-bb40-21337fa9269d	Gésier de dinde émincé 50%, graisse de canard ...	Gésier de dinde émincé 50%, graisse de canard...
13980d31-9002-457d-8d49-b451f08f473c	Edulcorants sorbitol, isomalt, sirop de maltit...	Edulcorants sorbitol, isomalt, sirop de maltit...
74297717-3fa8-4aed-95cc-e873c1a6157	sucres, amidon modifié, LACTOSÉRUM en poudre, d...	Z16005 / 002\sucres, amidon modifié, LACTOSÉRUM...

On constitue une table pour intégration dans le rapport :

```
[18]: with pd.option_context("max_colwidth", 100000):
      tex_str = (
          sub_sample.sample(20, random_state=41)
          .replace(r'\s*$', np.nan, regex=True)
          .to_latex(index=False,
                    index_names=False,
                    column_format='p{7cm}p{7cm}',
                    na_rep='<rien>',
                    longtable=True,
                    header=["Liste d'ingrédients cible", "Liste d'ingrédients prédite"],
                    label='tbl:GT_prediction_sample',
                    caption="Extrait des résultats de la prédiction",
                    )
          .replace(r'\textbackslash n', r' \newline ')
          .replace(r'\\', r'\\ \hline')
      )

# with open(Path('.') / 'tbls' / 'GT_prediction_sample.tex', 'w') as file:
#     file.write(tex_str)
```