Construction de la ground truth

Pierre Massé

June 11, 2020

1 Constitution de l'échantillon de données étiquetées

L'objet de ce notebook est de produire un échantillon données du PIM, avec les fiches techniques associées. Elles seront ensuite associées manuellement à la liste d'ingrédients qu'elles contiennent.

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 diplay
import os
import pandas as pd
pd.options.display.min_rows = 6
pd.options.display.width=108
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline

from src.pimapi import Requester
from src.pimest import ContentGetter, PDFContentParser
```

[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 Récupération des données, et de la présences de fiches techniques

Pour constituer l'échantillon, on va d'abord extraire quelques informations du PIM, et particulièrement le type de produit. On récupèrera aussi le fait que les produits ont ou non une fiche technique fournisseur associée.

[4]:
```
requester = Requester('prd')
# Let's fetch the full content of PIM system
requester.fetch_all_from_PIM()
requester.result
```

```
Done
```

[4]:
```
[<Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>]
```

[5]:
```
mapping = {'uid': 'uid',
           'designation': 'title',
           'state': 'state',
           'ingredients': 'properties.pprodc:ingredientsList',
           'type': 'properties.pprodtop:typeOfProduct'}
df = requester.file_report_from_result(mapping=mapping, index='uid') # , record_path='entries')
df.sample(5)
```

[5]:

uid	designation	state	ingredients	type	has_supplierdatasheet	has_supplierlabel
f2fc9f7a-ff2f-4502-a00e-7304fb7e6ee0	Mayonnaise allégée en seau 5,1 kg VALTONIA	product.validate	Eau, huile de colza 25 %, vinaigre, jaune d'OE...	grocery	False	False
3ec9cc99-7ba2-449d-af c7-b22e465a192a	MARJOLAINE 1KG	product.waiting.sending.supplier	None	grocery	False	False
ecbe45b9-070f-4086-90f4-fa3cc65a40e4	TORK LINGETTES IMPRÉGNÉES POUR NETTOYAGE DE SU...	product.validate	None	chemistry	True	True
a87747c6-e570-4a4e-8759-b7d400451436	Barquette charcutière 150 cc en sachet de 500 ...	product.controlAnoMinor	None	hygiene	True	True
afee12c7-177e-4a68-9539-8cbb68442503	DESTR D'ODEURS AIR&TEXTILES 750CCX6 DESODOR U2	product.waiting.supplier.validation	None	chemistry	False	False

1.2 Constitution de l'échantillon

On va constituer l'échantillon en appliquant les règles suivantes : - on construit un échantillon de 500 produits - on conserve les produits de type Epicerie et Boisson non alcoolisée - on conserve les produits qui possèdent une fiche technique fournisseur - on fait un échantillon stratifié par type de produit (Epicerie / Boisson)

[6]:
```
filtered_df = df.loc[(df.type.isin(['grocery', 'nonAlcoholicDrink']))
                     & (df.has_supplierdatasheet)]
train, ground_truth_df = train_test_split(filtered_df,
                                          test_size=500,
                                          random_state=42,
                                          stratify=filtered_df.type)
ground_truth_df.sample(5)
```

[6]:

uid	designation	state	ingredients	type	has_supplierdatasheet	has_supplierlabel
2a13382a-384d-4e50-8eef-2736698acef8	Penne Rigate sans-gluten en boîte 400 g BARILLA	product.waiting.supplier.validation	Farine de maïs blanc (60%), farine de maïs jau...	grocery	True	False
1806540a-33b7-4d7d-a045-943ed8b48799	Spécialité pomme-biscuit en coupelle 100 g CHA...	product.validate	Purée de pommes 95%, poudre de biscuit 4,2% (...	grocery	True	True
e1fe5f4b-9118-49a6-b02c-71b80407a7b9	Safran en poudre en boîte 10 g GANDOM	product.waiting.supplier.validation	Safran (Crocus Sativus Linnae)	grocery	True	False
3a4e72bb-b268-4c6b-9a51-48b78bdec4f2	GRAINE DE COURGE	product.validate	Graine de courge	grocery	True	True
8c147f0a-2388-4b1a-ac4a-51febc602278	Lasagnette à l'ancienne aux 7 œufs en sac 3 kg...	product.waiting.supplier.validation	Semoule de BLE dur de qualité supérieure, œUF...	grocery	True	False

Remarque : malgré l'utilisation d'un random_state fixé, l'échantillon généré n'est pas toujours le même à chaque exécution. En effet, comme la liste de produits varie au fil du temps (nouveaux référencements, périmètre des filtres qui change), le résultat du train_test_split peut varier.

Il s'agit ici seulement d'illustrer la méthode utilisée.

1.3 Export des pièces jointes du PIM et constitution du fichier d'étiquettes

On exporte ensuite le contenu du PIM sur le disque, afin d'avoir les fiches techniques simplement à disposition.

Remarque : des lignes dans ce paragraphe pour ne pas télécharger à nouveau les pièces jointes ni écraser le résultat de l'étiquetage manuel.

[ ]:
```
requester.fetch_list_from_PIM(ground_truth_df.index, batch_size=20)
# requester.dump_data_from_result(update_directory=False, root_path=os.path.join('.', 'ground_truth_to_del'))
# requester.dump_files_from_result(update_directory=False, root_path=os.path.join('.', 'ground_truth_to_del'))
```

On exporte également au format csv les uids des produits et les libellés associés (pour s'assurer qu'il n'y a pas eu de confusion lorsqu'on lit une fiche technique).

[ ]:
```
# ground_truth_df['designation'].to_csv(os.path.join('.', 'ground_truth_to_del', 'uid.csv'),
#                                       header=True,
#                                       encoding='utf-8-sig')
```

On teste également la possibilité de recharger les données depuis le fichier csv, une fois qu'il a été renseigné à la main dans excel.

[ ]:
```
# pd.read_csv(os.path.join('..', '..', 'ground_truth', 'manually_labelled_ground_truth.csv'),
#             sep=';',
#             encoding='latin-1',
#             index_col='uid')
```

1.4 Résultat de l'étiquetage manuel

Le résultat de l'étiquetage manuel est le suivant :

[7]:
```
df_gt = pd.read_csv(os.path.join('..', '..', 'ground_truth', 'manually_labelled_ground_truth.csv'),
                    sep=';',
                    encoding='latin-1',
                    index_col='uid')

def to_latex_newline(text):
    return(text.replace('\n', '  '))
```

```
with pd.option_context("max_colwidth", 1000):
    pass
#     print(df_gt)
#     df_gt.to_latex(
#                    Path('..') / 'tbls' / 'ground_truth.tex',
#                    index=False,
#                    index_names=False,
#                    column_format='p{5cm}p{10cm}',
#                    formatters=[to_latex_newline, to_latex_newline],
#                    longtable=True,
#                    na_rep='-',
#                    escape=True,
#                    )
df_gt.sample(5)
```

[7]:

uid	designation	ingredients
83dc5272-5e87-47b7-bd06-271bbac620a4	Flocon d'érable en sachet 170 g COULEUR QUEBEC	sirop d'érable pur à 100 %.
bbddc4ed-6d16-475c-ace1-851c8b32d28b	DEMI POIRES WILLIAMS AU SIROP LÉGER	NaN
a3d51821-275c-4471-8df4-b1fa1efede25	Purée d'épinard sans sel ajouté en sachet 1 kg...	Pommes de terre 59,5 % - Epinards 40 % - Amido...
6dfae8fd-6111-4a57-862e-c20a39a195e0	Pain de mie sans croûte en tranches en paquet ...	Farine de BLÉ 63%, eau, sucre, huile de colza,...
1678fd52-dc4b-4818-81de-b9c1581dc272	Spécialité pomme-abricot en boîte 5/1 VALADE E...	Pommes 78%, purée d'abricots à base de concent...

1.5 Comparatif entre les données étiquetées et le contenu du PIM

On peut comparer le contenu des listes d'ingrédients du PIM et les données étiquetées.

[8]:
```
requester = Requester('prd')
requester.fetch_all_from_PIM()
requester.result
```

Done

[8]:
```
[<Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>,
 <Response [200]>]
```

On récupère le contenu du PIM

[9]:
```
df = requester.result_to_dataframe()
pim_ds = df['properties.pprodc:ingredientsList']
pim_ds.sample(5)
```

[9]:
```
uid
e061a124-b312-4e5b-8311-d8f374ce0c01    80% Bolets jaunes Suillus Luteus, 20% Cèpes Bo…
f0a62940-5055-4433-83d0-8becdf6561e3    Sirop de glucose, sucre, eau, gélatine, acidif…
7b64e7a7-8c7c-45e3-80e9-48fdc6d44c64                                               None
115bf599-c6fa-40f7-ac08-622901756d0c                       Haricots verts, eau, sel
1afa5387-e2e3-4fc2-b991-b896f7feacc9                                               None
Name: properties.pprodc:ingredientsList, dtype: object
```

On charge le csv des données étiquetées :

[10]:
```
df_gt = pd.read_csv(os.path.join('..', '..', 'ground_truth', 'manually_labelled_ground_truth.csv'),
                    sep=';',
                    encoding='latin-1',
                    index_col='uid')
df_gt.sample(5)
```

[10]:

uid	designation	ingredients
d39f16bc-29f8-40b9-9d56-43295bfd5961	FLOWPACK PATAREV HIPPOPOTAMUS	PÂTE À MÂCHER ACIDE, AROMATISÉE : GOÛT FRAMBOI...
8097a8a8-86c0-4f9a-8c75-6d825a979e8c	Sucre cristal en sac 5 kg DADDY	NaN
4f83306f-66de-4545-9b12-7790b57b61ae	Nappage miroir neutre en seau 7 kg ANCEL	Sirop de glucose, sucre, eau, stabilisants (E4...
93e5d2af-10c5-4853-a437-b013673310cb	Riz long de Camargue IGP en sac 5 kg CANAVERE	NaN
5cb7f05a-3b2c-440e-af0d-01843fb38cbf	Assortiment de Malabar magic blue 3 parfums en...	Sucre, Gomme base, Sirop de glucose, Acidifian...

Comme l'index de la series des données issues du PIM, et du dataframe de la ground truth est le même (l'uid du produit), on peut faire très simplement la jointure via la méthode join :

[11]:
```
merged = (df_gt.join(pim_ds)
               .rename({'ingredients': 'Ingrédients de la ground truth',
                        'properties.pprodc:ingredientsList': 'Ingrédients du PIM'},
                       axis=1)
         )
merged.sample(5)
```

[11]:

uid	designation	Ingrédients de la ground truth	Ingrédients du PIM
1de02b1c-f17e-4d46-b90f-3a6c37ecf6aa	AMANDES DECORTIQUEES GRILLEES SANS SEL	AMANDES décortiquées	AMANDES grillées
c2ef743e-f3f2-4e8a-aab0-1e6cbeb71666	Gâteau aux céréales et aux graines de tourneso...	Farine de BLÉ 20% - Huile de colza - OEUFS ent...	Huile de colza - Farine de BLÉ 19,5% - OEUFS ...
ab48a1ed-7a3d-4686-bb6d-ab4f367cada8	Macaroni en sachet 500 g PANZANI	- 100% Semoule de BLE dur de qualité supérieur...	100% Semoule de BLE dur de qualité supérieure
5d24bc08-ff76-4ebb-800a-55aa3a6dc76d	Boissons énergisante en canette 47,3 cl RED BULL	eau gazéifiée, saccharose, glucose, correcteur...	Eau gazéifiée, saccharose, glucose, acidifiant...
70500268-802d-4211-93ba-9edbf6e0e7a3	COLIS KERMESSE 2019 A.P " 22+8"	TAGADA\nsucre; sirop de glucose; gélatine; aci...	GADA: sucre; sirop de glucose; gélatine; acid...

On peut compter les égalités strictes entre les ingrédients du PIM et ceux de la ground truth :

[12]:
```
merged['equals'] = (merged['Ingrédients du PIM'] == merged['Ingrédients de la ground truth'])
merged['equals'].value_counts()
```

[12]:
```
False    452
True      48
Name: equals, dtype: int64
```

Seules 50 listes d'ingrédients sont strictement identiques. Si on compare les listes qui ne le sont pas, on obtient :

[13]:
```
diff = merged.loc[~merged['equals'], ['Ingrédients du PIM', 'Ingrédients de la ground truth']]
for i in range(6):
    print('++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++')
    print('Issu du PIM :')
    print(diff.iloc[i].loc['Ingrédients du PIM'])
    print('--------------------------------------------------')
    print('Issu de la ground truth :')
    print(diff.iloc[i].loc['Ingrédients de la ground truth'])
    print('++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n')
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Issu du PIM :
Farine de BLE T65, eau, levure,  huile de colza, sel, vinaigre de cidre, assaisonnement poudre de curry,
agent de traitement de la farine : acide ascorbique, émulsifiant : E471
--------------------------------------------------
Issu de la ground truth :
Farine de blé T65, eau, levure, vinaigre de cidre, huile de colza, assaisonnement poudre de curry, sel,
acide ascorbique, émulsifiant : E471.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Issu du PIM :
100% Semoule de BLE dur de qualité supérieure
--------------------------------------------------
Issu de la ground truth :
- 100% Semoule de BLE dur de qualité supérieure
- Contient du gluten
Si le numéro de lot contient la lettre N : peu contenir de l'oeuf
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Issu du PIM :
Fève de tonka, taux de coumarine compris entre 1 et 3,5%
--------------------------------------------------
Issu de la ground truth :
fève de tonka (graines ridées de 25 à 50mm de long)
Taux de coumarine compris entre 1 et 3,5 %
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Issu du PIM :
Aubergine 60,5% (aubergine, huile de tournesol), eau, oignon, huile de tournesol, jus de citron, concentré
de tomate, huile d'olive vierge extra 2%, ail, sel, persil, basilic, poivre, thym, romarin.
```

```
--------------------------------------------------
Issu de la ground truth :
Aubergine 60,5% (aubergine, huile de tournesol), eau, oignon, huile de tournesol, jus de citron, concentré
de tomate, huile d'olive vierge extra (2%), ail, sel, persil, basilic, poivre, thym, romarin.
++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++
Issu du PIM :
Ingrédients : Myrtille Cassis : Fruits (myrtilles 41%, cassis 9%), sucre, sucre roux de canne, jus de
citrons concentré, gélifiant : pectines de fruits. Fraise Groseille : Fruits (fraises 27 %, groseilles 23
%), sucre, sucre roux de canne, jus de citrons concentré, gélifiant : pectines de fruits. Abricot Pêche :
Fruits (abricots 34%, pêches 16%), sucre, sucre roux de canne, jus de citrons concentré, gélifiant :
pectines de fruits. Orange Douce Mandarine : Fruits (oranges douces 37%, mandarines 3%), sucre, sucre roux
de canne, jus de citrons concentré, gélifiant : pectines de fruits.
--------------------------------------------------
Issu de la ground truth :
Confiture de myrtilles et de cassis
fruits (myrtilles 41%, cassis 9%), sucre, sucre roux de canne, jus de citrons concentré, gélifiant : pectine
de fruits.
Confiture de fraises et de groseilles
fruits (fraises 27 %, groseilles 23 %), sucre, sucre roux de canne, jus de citrons concentré, gélifiant :
pectine de fruits.
Confiture d'abricots et de pêches
fruits (abricots 34%, pêches 16%), sucre, sucre roux de canne, jus de citrons concentré, gélifiant : pectine
de fruits.
Marmelade d'oranges douces et de mandarines
fruits (oranges douces 37%, mandarines 3%), sucre, sucre roux de canne, jus de citrons
concentré, gélifiant : pectine de fruits.
++++++++++++++++++++++++++++++++++++++++++++++++++

++++++++++++++++++++++++++++++++++++++++++++++++++
Issu du PIM :
Pommes 95%, sirop de glucose-fructose, arôme, antioxydant : acide ascorbique
--------------------------------------------------
Issu de la ground truth :
Pommes 95%, sirop de glucose-fructose, arôme, antioxydant: acide ascorbique.
++++++++++++++++++++++++++++++++++++++++++++++++++
```

On peut sortir un tableau des données en écart, de manière basique :

[14]:
```
with pd.option_context("max_colwidth", 100000):
    tex_str = (
    diff.sample(10, random_state=44)
        .to_latex(index=False,
                  index_names=False,
                  column_format='p{7cm}p{7cm}',
                  na_rep='-',
                 )
        .replace(r'\textbackslash n', '\\newline ')
    )
#     print(tex_str)

# with open(Path('..') / 'tbls' / 'ingredient_comparison.tex', 'w') as file:
#     file.write(tex_str)
diff.sample(10, random_state=44)
```

[14]:

uid	Ingrédients du PIM	Ingrédients de la ground truth
bf633f9f-a89a-499b-afb8-1b874a477b08	Champignons, eau, sel , acidifiant : acide cit...	champignons, eau, sel, acidifiant : acide citr...
84d5c32f-92d0-49c9-9151-d1fd65238a2a	LAIT écrémé	Lait écrémé.
15d6958c-025e-43a6-9f3b-a0d923a61c3f	Pommes en tranches (43%), pêches en tranches (...	Pommes en tranches (35 à 56%), pêches en tranc...
eeca38ed-ff9b-467f-874d-298a350bd6c5	Pâtes alimentaires de semoule de BLÉ dur de qu...	SEMOULE DE BLE' DUR de qualité supérieure
6db330c3-26d0-4a46-93a5-74e704b107ff	Ecorce de citron (57%), sirop de glucose-fruct...	NaN
b70ed045-57ec-497d-bf18-af14fbbbe955	BLE dur entier précuit	NaN
e67341d8-350f-46f4-9154-4dbbb8035621	Sucre roux de canne*°(64%), amidon de maïs*, p...	Sucre roux de canne*° (64%), amidon de maïs*, ...
f9af1c71-59dd-4d11-8938-aa726ecffe6c	Eau, huile de tournesol, miel 10%, moutarde à ...	Eau, huile de tournesol, miel 10%, moutarde à ...
93fb1748-efa5-4679-b67e-51ff121c69e8	sucre de canne, eau, jus de mirabelle à base d...	SUCRE DE CANNE, EAU, JUS DE MIRABELLE A BASE D...
f9f2c425-07cd-43ef-aabe-ec777e89a6e7	Sucre 49,0%, NOISETTES 25,0%, AMANDES 25,0%, é...	sucre 49,0%; noisettes 25,0%; amandes 25,0%; é...

1.6 Analyse du contenu des pièces jointes téléchargées

On peut faire une estimation des pièces jointes dont les textes sont extractibles. On commence simplement par lister les pièces jointes relatives à la ground truth qui ont été téléchargées.

[15]:
```
p = Path('.') / 'ground_truth_to_del'
```

[19]:
```
files_df = pd.DataFrame(list(p.glob('**/*.pdf')), columns=['path'])
files_df['type'] = files_df['path'].apply(lambda x: x.name).apply(lambda x: x.split('.')[0])
files_df['uid'] = files_df['path'].apply(lambda x: x.parent.name)
files_df.set_index('uid', inplace=True)
files_df.sample(5)
```

[19]:

uid	path	type
ed969c94-33a2-4a82-bc84-0d4adc908f5c	ground_truth_to_del/ed969c94-33a2-4a82-bc84-0d...	FTF
244e14b8-8291-4315-8ca8-53fa85cf23f6	ground_truth_to_del/244e14b8-8291-4315-8ca8-53...	FTF
f42e19ae-d433-410d-a28d-ca01127b0ded	ground_truth_to_del/f42e19ae-d433-410d-a28d-ca...	FTF
57877d62-ace0-44ad-81bf-ed63b7a37877	ground_truth_to_del/57877d62-ace0-44ad-81bf-ed...	FTF
194419d0-d9f2-4799-81ac-d9e3aa77fd27	ground_truth_to_del/194419d0-d9f2-4799-81ac-d9...	FTF

On utilise les transformateur du module pimest pour récupérer le contenu de ces fichiers dans le dataframe.

[ ]:
```
transformer = make_pipeline(ContentGetter(), PDFContentParser())
files_df = transformer.fit_transform(files_df)
files_df
```

[ ]:
```
files_df['empty'] = (files_df['text'].apply(lambda x: x.strip()) == '')
files_df.sample(5)
```

[ ]:
```
(files_df.pivot_table(values='empty',
                      index='type',
                      aggfunc=['sum', 'count', 'mean'],
                     )
         .swaplevel(axis=1)
         .rename({'empty': 'Fichiers "vides"',
                  'sum': 'Nombre de fichiers vides',
                  'count': 'Nombre total de fichiers',
                  'mean': 'Taux de vides',
                 }, axis=1)
          .rename({'Etiquette': 'Etiquettes',
                   'FTF': 'Fiches techniques',
                  })

          .to_latex(
                    #Path('..') / 'tbls' / 'empty_attached_files.tex',
                    column_format='lccc',
                    bold_rows=True,
                    index_names=False,
                    formatters=[lambda x: str(int(x)),
                                lambda x: str(int(x)),
                                lambda x: f'{int(x * 100):d}%',
                               ]
                   )
)
pass
```