03_linking_trials_with_articles

March 16, 2025

[8]:
```python
import pandas as pd
pubmed = pd.read_csv('data/all_pubmed.csv')
pubmed.head()
```

[8]:
```
   pubmed_id                                              title  \
0   40073121  Targeting the NPY/NPY1R signaling axis in muta…
1   40069621  The value of preoperative RDW for post-pancrea…
2   40069616  Protocol of the IMPACT study: randomized, mult…
3   40066089  Association between human leukocyte antigen E …
4   40065459  Oncological and Survival Endpoints in Cancer C…

                                            keywords  \
0                                                NaN
1  Pancreatic ductal adenocarcinoma, Post-pancrea…
2  Atezolizumab, Bevacizumab, Conversion, Hepatoc…
3  HLA-E, cancer, human leukocyte antigen, immuno…
4  adverse events, cachexia, cancer, clinical tri…

                                      journal  \
0                           Science advances
1                                 BMC cancer
2                                 BMC cancer
3                       Frontiers in oncology
4  Journal of cachexia, sarcopenia and muscle

                                            abstract methods  \
0  Pancreatic cancer (PC) is a highly metastatic …     NaN
1  Pancreatic ductal adenocarcinoma (PDAC) is a h…     NaN
2  Atezolizumab plus bevacizumab is recommended a…     NaN
3  Immunotherapy has gained momentum with the dis…     NaN
4  In patients receiving anti-cancer treatment, c…     NaN

                                             results  \
0                                                NaN
1  A total of 2268 patients were analyzed. We fou…
2                                                NaN
3  After screening 657 articles, 11 studies were …
```

```
4  Fifty-seven trials were eligible, totalling 97…

                                         conclusions publication_date  \
0                                                NaN       2025-03-12
1  The preoperative RDW may be a useful marker fo…       2025-03-12
2                                                NaN       2025-03-12
3  This systematic review highlights that HLA-E e…       2025-03-11
4  In CC trials, oncological endpoints were mostl…       2025-03-11

                       category
0  Pubmed_Pancreatic_Cancer.csv
1  Pubmed_Pancreatic_Cancer.csv
2  Pubmed_Pancreatic_Cancer.csv
3  Pubmed_Pancreatic_Cancer.csv
4  Pubmed_Pancreatic_Cancer.csv
```

[9]:
```
import pandas as pd
import re
import json

# Function to extract NCT IDs safely
def extract_nct_ids_from_abstract(abstract):
    """
    Extract all NCT IDs from the abstract using regex.
    Ensure that the input is a string before applying regex.
    """
    if not isinstance(abstract, str):  # Ensure it's a valid string
        return []
    return re.findall(r'NCT\d+', abstract)

# Function to link PubMed articles to Clinical Trials
def link_pubmed_to_trials(pubmed_df):
    """
    Link PubMed articles to Clinical Trials using NCT IDs extracted from the␣
 ↪abstracts.
    """
    data = []

    for index, row in pubmed_df.iterrows():
        pubmed_id = row['pubmed_id']
        abstract = row.get('abstract', '')  # Get abstract safely, default to␣
 ↪empty string
        nct_ids = extract_nct_ids_from_abstract(abstract)

        if nct_ids:  # Only add if there are valid NCT IDs
            data.append({"PubMed_ID": pubmed_id, "NCT_IDs": nct_ids})
    return data

# Extracting and saving to JSON
linked_data = link_pubmed_to_trials(pubmed)

# Saving results to JSON
output_file = "data/linked_pubmed_nct_ids.json"
with open(output_file, 'w') as json_file:
    json.dump(linked_data, json_file, indent=4)

print(f"Data saved in {output_file}")
```

```
Data saved in data/linked_pubmed_nct_ids.json
```

[10]:
```
import pandas as pd
import json

# pd.read_csv('data/all_diseas_processed.csv')
all_disease = pd.read_csv('data/CT_all_common_disease_processed.csv')
# print(all_disease.head())

pubmed_links = json.load(open('data/linked_pubmed_nct_ids.json'))
# print(pubmed_links[:5])





# Create a mapping of NCT ID to PubMed ID
nct_to_pubmed = {}

for entry in pubmed_links:
    pubmed_id = str(entry["PubMed_ID"])  # Ensure string format
    for nct_id in entry["NCT_IDs"]:
        nct_to_pubmed[nct_id] = pubmed_id

# Sample all_disease DataFrame
# all_disease = pd.DataFrame({
#     "NCT ID": ["NCT06088706", "NCT02871856", "NCT12345678", "NCT05622630"]
# })

# Map NCT ID to PubMed ID
all_disease["Associated Article ID"] = all_disease["NCT ID"].map(nct_to_pubmed).
 ↪fillna("")

# Create binary column indicating association
all_disease["Associated Article?"] = all_disease["Associated Article ID"].
 ↪apply(lambda x: "YES" if x else "NO")
```

```
# Display result
# print(all_disease)

print("Count of all common diseases:", len(all_disease))
print("Number of trials with associated articles:", all_disease['Associated␣
↪Article?'].value_counts()['YES'])
print("Number of trials not having any associated articles:",␣
↪all_disease['Associated Article?'].value_counts()['NO'])
```

```
Count of all common diseases: 14118
Number of trials with associated articles: 1137
Number of trials not having any associated articles: 12981
```

[11]:
```
import pandas as pd
import json

# pd.read_csv('data/all_diseas_processed.csv')
all_disease = pd.read_csv('data/CT_all_rare_disease_processed.csv')
# print(all_disease.head())

pubmed_links = json.load(open('data/linked_pubmed_nct_ids.json'))
# print(pubmed_links[:5])




# Create a mapping of NCT ID to PubMed ID
nct_to_pubmed = {}

for entry in pubmed_links:
    pubmed_id = str(entry["PubMed_ID"])  # Ensure string format
    for nct_id in entry["NCT_IDs"]:
        nct_to_pubmed[nct_id] = pubmed_id

# Sample all_disease DataFrame
# all_disease = pd.DataFrame({
#     "NCT ID": ["NCT06088706", "NCT02871856", "NCT12345678", "NCT05622630"]
# })

# Map NCT ID to PubMed ID
all_disease["Associated Article ID"] = all_disease["NCT ID"].map(nct_to_pubmed).
↪fillna("")

# Create binary column indicating association
all_disease["Associated Article?"] = all_disease["Associated Article ID"].
↪apply(lambda x: "YES" if x else "NO")
```

```python
# Display result
# print(all_disease)

print("Count of all rare diseases:", len(all_disease))
print("Number of trials with associated articles:", all_disease['Associated␣
↪Article?'].value_counts()['YES'])
print("Number of trials not having any associated articles:",␣
↪all_disease['Associated Article?'].value_counts()['NO'])
```

```
Count of all rare diseases: 3648
Number of trials with associated articles: 422
Number of trials not having any associated articles: 3226
```