Beyond twitter - Exploring new social networks for digital disease detection

Case study: France

Heiner Atze, PhD

```
import pandas as pd
import matplotlib.pyplot as plt
import langdetect

from multiprocessing import Pool
```

Introduction

- Elon Musk events
- history of Bluesky
- technical details (decentralized)

Methods

Datasets

Official data

Sentinelles

```
sent = pd.read_csv('./data/inc-25-PAY-ds2.csv')
```

WHO

```
flunetfr = pd.read_csv("./data/flunet.csv")\
    .query("COUNTRY_CODE=='FRA'")
flunetfr['ISO_WEEKSTARTDATE'] = pd.to_datetime(flunetfr.ISO_WEEKSTARTDATE)
fluidfr = pd.read_csv("./data/fluid.csv")\
    .query("COUNTRY_CODE=='FRA'")
fluidfr['ISO_WEEKSTARTDATE'] = pd.to_datetime(fluidfr.ISO_WEEKSTARTDATE)
```

```
/tmp/ipykernel_2065680/2540891737.py:4: DtypeWarning: Columns (28,29) have mixed types. Spec
```

```
fluidfr = pd.read_csv("./data/fluid.csv")\
```

Bluesky messages

```
grippe = pd.concat(
    [
        pd.read_csv(f) for f in (
            "./data/Bluesky test grippe.csv",
            "./data/Bluesky test grippe end 24-11-11.csv"
            )
        ]
).drop_duplicates().dropna(subset = 'text').reset_index(drop=True)
grippe['date'] = pd.to_datetime(grippe['date'], format = 'mixed')

rhume = pd.read_csv(
    "./data/Bluesky rhume.csv",
)
rhume['date'] = pd.to_datetime(rhume['date'], format = 'mixed')
```

Surveillance data

Bluesky messages

Bluesky datasets were extracted the APIs available at Gruzd (2025).

Keyword list

- grippe
- rhume

Language detection

Language detection to disambiguate French from German tweets (grippe) was performed using the `langdetect` [ref] module.

```
def detect_language(msg: str) -> str:
    try:
        return langdetect.detect(msg)
    except:
        return ""
```

```
with Pool(8) as p:
    grippe['lang'] = p.map(detect_language, grippe.text)
```

```
grippe_fr_tw = grippe.query("lang=='fr' & ~(text.str.contains('aviaire'))").set_index('date'
filter_idx = grippe_fr_tw.index

rhume_tw = rhume.set_index('date').resample('1w').count().text.loc[filter_idx]
```

```
fluidfr = fluidfr.set_index("ISO_WEEKSTARTDATE").resample("1w").sum().loc[filter_idx]
flunetfr = flunetfr.set_index("ISO_WEEKSTARTDATE").resample("1w").sum().loc[filter_idx]
```

```
/tmp/ipykernel_2065680/700646525.py:1: FutureWarning: 'w' is deprecated and will be removed
  grippe_fr_tw = grippe.query("lang=='fr' & ~(text.str.contains('aviaire'))").set_index('dat
/tmp/ipykernel_2065680/700646525.py:1: FutureWarning: 'w' is deprecated and will be removed
  grippe_fr_tw = grippe.query("lang=='fr' & ~(text.str.contains('aviaire'))").set_index('dat
/tmp/ipykernel_2065680/700646525.py:4: FutureWarning: 'w' is deprecated and will be removed
  rhume_tw = rhume.set_index('date').resample('1w').count().text.loc[filter_idx]
/tmp/ipykernel_2065680/700646525.py:5: FutureWarning: 'w' is deprecated and will be removed
  fluidfr = fluidfr.set_index("ISO_WEEKSTARTDATE").resample("1w").sum().loc[filter_idx]
/tmp/ipykernel_2065680/700646525.py:6: FutureWarning: 'w' is deprecated and will be removed
  flunetfr = flunetfr.set_index("ISO_WEEKSTARTDATE").resample("1w").sum().loc[filter_idx]
```

```
def rolling_std(data, window: int):
    tmp = np.zeros_like(data)
    w_start = 0
    w_end = window - 1

    for i in range(data.shape[0] - window):
        sl = data[w_start:w_end,:]
        mu = sl.mean(axis = 0)
        sigma = sl.std(axis = 0) or 1
        tmp[w_end+1,:][:] = np.divide(np.subtract(data[w_end+1,:], mu), sigma)

        w_start += 1
        w_end += 1

    return tmp
```

```
import numpy as np
```

```
df = pd.DataFrame()
df.index = filter_idx

df['grippe'] = grippe_fr_tw
df['grippe_std'] = rolling_std(df[['grippe']].values, 7)
df['rhume'] = rhume_tw
df['ili_cases'] = fluidfr.ILI_CASE
df['inf_vir'] = flunetfr.INF_ALL
df['inf_vir_std'] = rolling_std(flunetfr[['INF_ALL']].values, 7)
```

Results

Correlation analysis

Complete dataset

```
df.corr()
```

	grippe	grippe_std	rhume	ili_cases	inf_vir	inf_vir_std
grippe	1.000000	0.170576	0.569915	0.445713	0.249786	0.266103
grippe_std	0.170576	1.000000	0.146348	0.150384	0.073698	0.223127
rhume	0.569915	0.146348	1.000000	0.481225	0.259808	0.441175
ili_cases	0.445713	0.150384	0.481225	1.000000	0.878247	0.408153
inf_vir	0.249786	0.073698	0.259808	0.878247	1.000000	0.239532
inf_vir_std	0.266103	0.223127	0.441175	0.408153	0.239532	1.000000

Truncated dataset

```
df.loc['2024':].corr()
```

	grippe	grippe_std	rhume	ili_cases	inf_vir	inf_vir_std
grippe	1.000000	0.381794	0.473187	0.544890	0.473708	0.371637
grippe_std	0.381794	1.000000	0.223244	0.208414	0.128721	0.271536
rhume	0.473187	0.223244	1.000000	0.554001	0.443943	0.585296
ili_cases	0.544890	0.208414	0.554001	1.000000	0.936550	0.291056
inf_vir	0.473708	0.128721	0.443943	0.936550	1.000000	0.225330
inf_vir_std	0.371637	0.271536	0.585296	0.291056	0.225330	1.000000

Bibliograph

Gruzd, & Mai, A. 2025. “Communalytic: A No-Code Computational Social Science Research Tool for Studying Online Communities and Public Discourse on Social Media.” https://Communalytic.org.