Explode Synsets

1 Boilerplate

2 Imports

2.1 prod: NVM

```
from nvm import disp_df
from nvm import clean_str
from nvm.aux_str import CLEAN_STR_MAPPINGS_LARGE as maps0
from nvm.aux_str import REGEX_ABC_DASH_XYZ_ASTERISK as re0
from nvm.aux_pandas import fix_column_names
```

2.2 prod: Basics

```
import os
import pathlib
import numpy as np
import pandas as pd
import re
import json
import yaml
import srsly
import uuid
import random
import numbers
from collections import OrderedDict
from contextlib import ExitStack
import warnings
# warnings.warn("\nwarning")
from hashlib import md5
import humanfriendly as hf
import time
import datetime as dt
from pytz import timezone as tz
tz0 = tz("Europe/Berlin")
from glob import glob
from tqdm import tqdm
import logging
log0.info("DONE: basic imports")
```

2.3 prod: Extra imports and settings

```
from contexttimer import Timer
import textwrap

HOME = pathlib.Path.home()

tqdm.pandas()

import matplotlib
from matplotlib import pyplot as plt
# import seaborn as sns
# import plotly.graph_objects as go
# import plotly.express as px

# get_ipython().run_line_magic("matplotlib", "qt")
# get_ipython().run_line_magic("matplotlib", "inline")

with Timer() as elapsed:
    time.sleep(0.001)

log0.info(hf.format_timespan(elapsed.elapsed))

log0.info("DONE: extra imports and settings")
```

3 Extra Imports

3.1 prod: More extra imports and settings

```
log0.info("DONE: more extra imports and settings")
```

4 Process

4.1 prod: Load data

```
dir0 = "../../data/d0009_synsets-merged/"
dir0 = pathlib.Path(dir0)
# dir0.mkdir(mode=0o700, parents=True, exist_ok=True)
assert dir0.exists(), f"The data directory dir0={str(dir0)} not found!"

name0 = f"synsets-merged"
extn0 = ".pkl"

if0 = (dir0/name0).with_suffix(extn0)
log0.info(f"loading: {if0}...")
df0 = pd.read_pickle(if0)
log0.info(f"loading: {if0}... DONE")
```

```
df0["lemmas"] = df0["lemmas"].apply(lambda list0: [str(item0) for item0 in list0])
df0["target"] = df0.MEAN / 3
log0.info(f"{df0.target.mean() = }")
log0.info(f"{df0.target.min() = }")
log0.info(f"{df0.target.max() = }")
log0.info(f"{df0.target.std() = }")

log0.info(f"{df0.shape = }")
disp_df(df0.sample(n=8).sort_index())
```

```
I: loading: ../../data/d0009_synsets-merged/synsets-merged.pkl...
I: loading: ../../data/d0009_synsets-merged/synsets-merged.pkl... DONE
I: df0.target.mean() = 0.05918228939188339
I: df0.target.min() = -1.0
I: df0.target.max() = 0.9583333333333334
I: df0.target.std() = 0.3259159215888112
I: df0.shape = (9089, 9)

             TYPE             id0      MEAN       STD   CNT
105          src0  oewn-07042734-n  0.100000  0.316228  10.0  [musical theme, theme, melod
913          src0  oewn-08680308-n  0.000000  0.447214  11.0                [laboratory,
1531         src0  oewn-00844276-s  0.000000  1.632993   7.0
2529         src0  oewn-02562363-v  0.375000  1.187735   8.0
3815         src0  oewn-00604693-v  2.125000  0.991031   8.0          [train, develop, pr
4109         src0  oewn-00583425-n  0.125000  0.353553   8.0  [line, job, occupation, busi
4826         src0  oewn-01668036-a  1.428571  0.975900   7.0
7982  src0_negLem  oewn-00128351-n  1.625000  1.060660   8.0
```

4.2 Check lead

```
disp_df(
    df0[df0.lemmas.apply(lambda x: "lead" in x)],
    max_colwidth=222,
)
```

```
      TYPE              id0      MEAN       STD   CNT
162   src0  oewn-06664322-n  0.888889  0.927961   9.0  [tip, lead, wind, confidential info
773   src0  oewn-01259362-n  1.909091  0.943880  11.0
1185  src0  oewn-00773677-v  1.000000  1.414214  11.0
1416  src0  oewn-05164526-n  1.000000  0.755929   8.0
1677  src0  oewn-05835238-n  1.125000  0.991031   8.0
1993  src0  oewn-02003455-v  1.285714  0.487950   7.0
2046  src0  oewn-03610056-n  0.000000  0.000000   8.0         [lead, jumper cable, booste
2132  src0  oewn-02642040-v  0.666667  0.866025   9.0
2317  src0  oewn-14667645-n  0.250000  0.707107   8.0                              [lead,
2755  src0  oewn-02445109-v  1.857143  1.069045   7.0
3722  src0  oewn-02691775-v  0.888889  0.927961   9.0                              [run, ex
3854  src0  oewn-02693227-v  2.250000  0.462910   8.0
4420  src0  oewn-08609721-n  1.375000  1.187735   8.0
```

```
4530  src0  oewn-02003830-v  1.125000  0.834523   8.0                         [take, guide, l
5570  src0  oewn-02692313-v  0.625000  0.916125   8.0
5907  src0  oewn-01736802-v  2.250000  0.707107   8.0                                      [l
6253  src0  oewn-06281532-n  0.500000  1.069045   8.0
6559  src0  oewn-01258857-n  0.555556  0.726483   9.0
6661  src0  oewn-02561616-v  1.285714  0.951190   7.0                                 [lead,
```

4.3 Check fruitfully

```
disp_df(
    df0[df0.lemmas.apply(lambda x: "fruitfully" in x)],
    max_colwidth=222,
)
```

```
      TYPE              id0  MEAN      STD  CNT                                      lemmas
1732  src0  oewn-00215173-r   2.0  0.92582  8.0  [fruitfully, profitably, productively]  i
```

4.4 Check categories

```
print(f"{df0.lemmas.apply(len).mean()}")
print(f"{df0.lemmas.apply(len).std()}")
print(f"{df0.examples.apply(len).mean()}")
print(f"{df0.examples.apply(len).std()}")
```

```
2.03432720871383
1.6447636819370604
1.9046099680932995
1.2817403665854505
```

4.5 Check categories

```
df0.TYPE.value_counts()
```

```
TYPE
src0           6914
src0_negLem     872
ant2            436
ant2_negLem     436
ant0_negLem     431
Name: count, dtype: int64
```

4.6 Check examples

```
df0.examples.apply(lambda x: len(x)).value_counts().sort_index()
```

```
examples
1     4563
2     2541
3     1114
4      443
5      217
6      120
7       41
8       20
9       10
10      12
11       7
13       1
Name: count, dtype: int64
```

4.7 Cols

```
for col0 in df0.columns:
    print(f"    \"{col0}\",")
```

```
    "TYPE",
    "id0",
    "MEAN",
    "STD",
    "CNT",
    "lemmas",
    "definition",
    "examples",
    "target",
```

4.8 Copy source df

```
df1 = df0[df0.TYPE=="src0"].copy()
df1["definition"] = ""
df1["lemmas"] = [[]]*len(df1)

df2 = df0[df0.TYPE=="src0"].copy()
df4 = df0.copy()
df6 = df0.copy()
log0.info(f"{df2.shape = }")
log0.info(f"{df4.shape = }")
```

```
I: df2.shape = (6914, 9)
I: df4.shape = (9089, 9)
```

4.9 Explode examples in df1

```
df1a = df1.copy()
log0.info(f"raw: {df1a.shape = }")
df1a = df1a.explode(column=["examples"])
df1a["text"] = df1a["examples"]
log0.info(f"exp: {df1a.shape = }")
disp_df(df1a.sample(n=8).sort_index())
```

```
I: raw: df1a.shape = (6914, 9)
I: exp: df1a.shape = (12812, 10)

      TYPE              id0      MEAN       STD   CNT lemmas definition
556   src0  oewn-01531310-v  1.300000  0.948683  10.0     []             He finally could
1629  src0  oewn-00755631-s  1.333333  1.118034   9.0     []                       she rei
1871  src0  oewn-02751207-v  0.000000  0.500000   9.0     []
2018  src0  oewn-00430156-r  2.250000  0.886405   8.0     []             the federal gover
2899  src0  oewn-00969657-v  0.777778  0.666667   9.0     []                         publi
3840  src0  oewn-00558544-s -1.125000  0.834523   8.0     []
4050  src0  oewn-00342215-v  1.428571  0.975900   7.0     []
4894  src0  oewn-00796324-s  0.625000  0.916125   8.0     []                         overc
```

4.10 Explode lemmas in df2

```
df2a = df2.copy()
log0.info(f"raw: {df2a.shape = }")
df2a = df2a.explode(column=["lemmas"])
df2a["text"] = df2a.lemmas + ", " + df2a.definition
log0.info(f"exp: {df2a.shape = }")
disp_df(df2a.sample(n=8).sort_index())
```

```
I: raw: df2a.shape = (6914, 9)
I: exp: df2a.shape = (14813, 10)

      TYPE              id0      MEAN       STD   CNT       lemmas
663   src0  oewn-00582390-s -0.875000  1.246423   8.0       crying  conspicuously and outr
997   src0  oewn-01253673-v -1.000000  1.054093  10.0        worry                      to
2589  src0  oewn-00077383-n -1.000000  1.195229   8.0         trip    an unintentional but
2907  src0  oewn-00845580-s  1.750000  0.707107   8.0     emphatic
3813  src0  oewn-01688793-s -1.625000  0.916125   8.0  disoriented  having lost your beari
3992  src0  oewn-07224193-n  0.142857  1.214986   7.0    exception                grounds
6336  src0  oewn-06660952-n  0.375000  0.744024   8.0    testimony            something th
6817  src0  oewn-00032610-s  1.714286  0.951190   7.0    gymnastic
```

4.11 Explode examples in df2

```
df2b = df2.copy()
log0.info(f"raw: {df2b.shape = }")
df2b = df2b.explode(column=["examples"])
df2b["text"] = df2b.examples
log0.info(f"exp: {df2b.shape = }")
disp_df(df2b.sample(n=8).sort_index())
```

```
I: raw: df2b.shape = (6914, 9)
I: exp: df2b.shape = (12812, 10)
```

```
      TYPE             id0      MEAN       STD  CNT                        lemmas
472   src0  oewn-03977398-n  0.000000  0.000000  9.0                         [ply]  one o
1586  src0  oewn-01638779-s  0.375000  0.517549  8.0                    [official]
1800  src0  oewn-00440298-v  1.571429  0.975900  7.0  [speed, speed up, accelerate]
2960  src0  oewn-04707990-n  0.285714  0.755929  7.0   [coating, finishing, finish]  a dec
2970  src0  oewn-01399805-a  0.000000  0.000000  7.0                      [leaded]
4354  src0  oewn-01929647-v -1.500000  1.069045  8.0            [drift, err, stray]     wa
6403  src0  oewn-01653333-s -0.125000  0.353553  8.0                      [junior]   incl
6441  src0  oewn-05818587-n -0.555556  1.130388  9.0                [life, living]  the e
```

4.12 Explode lemmas in df4

```
df4a = df4.copy()
log0.info(f"raw: {df4a.shape = }")
df4a = df4a.explode(column=["lemmas"])
df4a["text"] = df4a.lemmas + ", " + df4a.definition
log0.info(f"exp: {df4a.shape = }")
disp_df(df4a.sample(n=8).sort_index())
```

```
I: raw: df4a.shape = (9089, 9)
I: exp: df4a.shape = (18490, 10)
```

```
             TYPE             id0      MEAN       STD  CNT          lemmas
1459         src0  oewn-00430425-n -0.833333  1.329160  6.0        escapism  an inclinatio
2796         src0  oewn-00891076-v  1.555556  0.726483  9.0       guarantee
4719         src0  oewn-04642461-n  1.250000  1.035098  8.0        activity  the trait of
5400         src0  oewn-00591299-v  1.250000  1.281740  8.0           catch  grasp with th
5429         src0  oewn-00814485-a  1.750000  1.164965  8.0           eager  having or sho
7942  src0_negLem  oewn-00782933-a  0.625000  0.916125  8.0  not indistinct  easy to perce
8142  src0_negLem  oewn-02101168-a  0.625000  0.916125  8.0    not insecure
8824  ant0_negLem  oewn-02486512-v  0.000000  1.732051  7.0  not legitimise
```

4.13 Explode examples in df4

```
df4b = df4.copy()
log0.info(f"raw: {df4b.shape = }")
df4b = df4b.explode(column=["examples"])
df4b["text"] = df4b.examples
log0.info(f"exp: {df4b.shape = }")
disp_df(df4b.sample(n=8).sort_index())
```

```
I: raw: df4b.shape = (9089, 9)
I: exp: df4b.shape = (17311, 10)
```

```
      TYPE              id0      MEAN       STD  CNT                                 lemmas
46    src0  oewn-04686906-n  0.125000  0.640870  8.0               [visage, countenance]
2068  src0  oewn-02235691-s  1.125000  1.246423  8.0                 [technical, expert]
2306  src0  oewn-00523831-v  0.000000  0.000000  7.0  [receive, get, incur, obtain, find]
3942  src0  oewn-00313654-s -0.714286  1.704336  7.0                [reckless, heedless]
4510  src0  oewn-01961388-v -0.142857  0.377964  7.0                         [sit, ride]
4640  src0  oewn-01632091-v -0.125000  1.642081  8.0                               [set]
6734  src0  oewn-02078305-v  0.666667  1.000000  9.0                [elude, evade, bilk]
7247  ant2  oewn-00635278-a -1.125000       NaN  NaN                  [incorrect, wrong]
```

4.14 Explode lemmas in df6

```
df6a = df6.copy()
log0.info(f"raw: {df6a.shape = }")
df6a = df6a.explode(column=["lemmas"])
df6a["text"] = df6a.lemmas + ", " + df6a.definition
log0.info(f"exp: {df6a.shape = }")
disp_df(df6a.sample(n=8).sort_index())
```

```
I: raw: df6a.shape = (9089, 9)
I: exp: df6a.shape = (18490, 10)
```

```
             TYPE              id0      MEAN       STD   CNT       lemmas
2188         src0  oewn-02108248-s -1.000000  1.290994   7.0        petty
2308         src0  oewn-00074163-r  0.250000  0.707107   8.0    primarily
3014         src0  oewn-02547977-v -0.111111  0.927961   9.0     abide by  act in accordan
3427         src0  oewn-00120604-v  0.142857  0.690066   7.0         have  cause to move;
4633         src0  oewn-00241051-n  1.750000  0.886405   8.0   foundation  the act of star
6349         src0  oewn-02386868-v  1.000000  1.069045   8.0       assume  take on titles,
7430  src0_negLem  oewn-01531310-v  1.300000  0.948683  10.0  not deposit           remove
8358  ant0_negLem  oewn-01394303-a -0.250000  0.462910   8.0    not large  limited or belo
```

4.15 Explode examples in df6

```
df6b = df6a.copy()  # WARN: CAUTION: taking df6a NOT df6
log0.info(f"raw: {df6b.shape = }")
df6b = df6b.explode(column=["examples"])
df6b["text"] = df6b.lemmas + "; " + df6b.definition + "; " + df6b.examples
log0.info(f"exp: {df6b.shape = }")
disp_df(df6b.sample(n=8).sort_index())
```

```
I: raw: df6b.shape = (18490, 10)
I: exp: df6b.shape = (41313, 10)
```

```
      TYPE              id0      MEAN       STD  CNT       lemmas
800   src0  oewn-02006442-v  0.111111  1.833333  9.0         fire       drive out or away
1619  src0  oewn-00305748-s  1.000000  1.264911  6.0        rough            violently ag
2622  src0  oewn-00005041-v  0.857143  1.214986  7.0      inspire
2829  src0  oewn-01644397-v  2.777778  0.440959  9.0   accomplish
3148  src0  oewn-04647089-n  0.250000  2.121320  8.0        rigor
4078  src0  oewn-02473075-s -1.111111  1.269296  9.0  questioning                 marked
5829  src0  oewn-04652076-n  1.625000  1.302470  8.0         zeal
5948  src0  oewn-01622528-s -0.125000  0.353553  8.0     manifest  clearly revealed to the
```

4.16 Concatenate

```python
cols0 = ["text", "target"]
df1x = df1a[cols0]
df2x = pd.concat([df2a[cols0], df2b[cols0]]).reset_index(drop=True)
df4x = pd.concat([df4a[cols0], df4b[cols0]]).reset_index(drop=True)
df6x = pd.concat([df6a[cols0], df6b[cols0]]).reset_index(drop=True)

log0.info(f"{df1x.shape = }")
log0.info(f"{df2x.shape = }")
log0.info(f"{df4x.shape = }")
log0.info(f"{df6x.shape = }")
disp_df(df1x.sample(n=8).sort_index(), max_colwidth=111)
disp_df(df2x.sample(n=8).sort_index(), max_colwidth=111)
disp_df(df4x.sample(n=8).sort_index(), max_colwidth=111)
disp_df(df6x.sample(n=8).sort_index(), max_colwidth=111)
```

```
I: df1x.shape = (12812, 2)
I: df2x.shape = (27625, 2)
I: df4x.shape = (35801, 2)
I: df6x.shape = (59803, 2)
```

```
                                                                   text    target
1858                             Who's running for treasurer this year?  0.416667
3312                                                a great work of art  0.458333
3388                                                      a sharp knife  0.208333
3634                         she fell to pieces after she lost her work -0.190476
3881                                           brads are headless nails -0.041667
4398  deductible losses on sales or exchanges of property are allowable  0.125000
5440                      The police car pursued the suspected attacker  0.523810
5498                                                    popular fiction  0.111111
```

```
2186                                                                                root,
3042                                                                                  firs
3310   trade, the commercial exchange (buying and selling on domestic or international mar
4014
12625                                                             employ, the state of bei
16703
23967                                                                          He spent th
25183                                                                           This adds
```

```
                                                                       text    target
509    big top, a canvas tent to house the audience at a circus performance -0.030303
2238                      organizational, of or relating to an organization  0.185185
8838                                       mystifying, of an obscure nature -0.041667
16205                               not lasting, not permanent; not lasting -0.285714
20678                                                  respiratory activity  0.222222
23724                                   bold settlers on some foreign shore  0.666667
28403                                                           gentle rain -0.458333
29629                                      Social relations impose courtesy  0.541667
```

```
9019                                                      superiority, displaying a sense
14587                                                  pull through, continue in existence
18850                                                      activate; put in motion or move
22564                                  rotter; a person who is deemed to be despicable or
33579                                                        wise; improperly forward or b
41189  tension; (psychology) a state of mental or emotional strain or suspense; he suffere
45651                                                                       seethe; boil v
48211                                                 head; to go or travel towards; We we
```

4.17 Save DATASETS (CAUTION)

```python
import pathlib
import csv
import datetime as dt
from pytz import timezone as tz
tz0 = tz("Europe/Berlin")

dir0 = "../../data/d0010_training-data/"
dir0 = pathlib.Path(dir0)
dir0.mkdir(mode=0o700, parents=True, exist_ok=True)
assert dir0.exists(), f"The data directory dir0={str(dir0)} was not found!"


now0 = []
pfx0 = ["ft0x"]
sfx0 = []

bfn0 = dir0/"_".join(pfx0+now0+sfx0).replace(".", "_")

xtn0 = ".pkl"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
df1x.to_pickle(ofn0)

xtn0 = ".csv"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
df1x.to_csv(ofn0, index=False, quoting=csv.QUOTE_NONNUMERIC)


now0 = []
```

```
pfx0 = ["ft1x"]
sfx0 = []

bfn0 = dir0/"_".join(pfx0+now0+sfx0).replace(".", "_")

xtn0 = ".pkl"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
df2x.to_pickle(ofn0)

xtn0 = ".csv"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
df2x.to_csv(ofn0, index=False, quoting=csv.QUOTE_NONNUMERIC)


now0 = []
pfx0 = ["ft2x"]
sfx0 = []

bfn0 = dir0/"_".join(pfx0+now0+sfx0).replace(".", "_")

xtn0 = ".pkl"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
df4x.to_pickle(ofn0)

xtn0 = ".csv"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
df4x.to_csv(ofn0, index=False, quoting=csv.QUOTE_NONNUMERIC)


now0 = []
pfx0 = ["ft3x"]
sfx0 = []

bfn0 = dir0/"_".join(pfx0+now0+sfx0).replace(".", "_")

xtn0 = ".pkl"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
df6x.to_pickle(ofn0)

xtn0 = ".csv"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
df6x.to_csv(ofn0, index=False, quoting=csv.QUOTE_NONNUMERIC)

log0.info("DONE")
```

```
I: saving: ../../data/d0010_training-data/ft0x.pkl...
I: saving: ../../data/d0010_training-data/ft0x.csv...
```

```
I: saving: ../../data/d0010_training-data/ft1x.pkl...
I: saving: ../../data/d0010_training-data/ft1x.csv...
I: saving: ../../data/d0010_training-data/ft2x.pkl...
I: saving: ../../data/d0010_training-data/ft2x.csv...
I: saving: ../../data/d0010_training-data/ft3x.pkl...
I: saving: ../../data/d0010_training-data/ft3x.csv...
I: DONE
```

4.18 Checkups

```
log0.info(f"{df0.target.mean() = }")
log0.info(f"{df0.target.std() = }")
log0.info(f"{df0.target.min() = }")
log0.info(f"{df0.target.max() = }")
```

```
I: df0.target.mean() = 0.05918228939188339
I: df0.target.std() = 0.3259159215888112
I: df0.target.min() = -1.0
I: df0.target.max() = 0.9583333333333334
```