Expand Patterns

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")
```

```
I: DONE: more extra imports and settings
```

4 Process

4.1 prod: Load data

```
dir0 = "../../data/d0004_sources-merged-and-deduped/"
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"merged"
extn0 = ".yaml"
if0 = (dir0/name0).with_suffix(extn0)
data0 = srsly.read_yaml(if0)
data0 = sorted(list(data0))
```

```
log0.info(f"{type(data0) = }")
log0.info(f"{type(data0[42]) = }")
log0.info(f"{data0[42] = }")
log0.info(f"{len(data0) = }")
```

```
I: type(data0) = <class 'list'>
I: type(data0[42]) = <class 'str'>
I: data0[42] = 'accura*'
I: len(data0) = 4618
```

4.2 prod: Load frequency rank data

```
dir2 = "../../data/d0000_word2vec-freq-ranks/"
dir2 = pathlib.Path(dir2)
# dir2.mkdir(mode=0o700, parents=True, exist_ok=True)
assert dir2.exists(), f"The data directory dir2={str(dir2)} not found!"

name0 = f"word2vec_freq_ranks_raw"
extn0 = ".jsonl"
if0 = (dir2/name0).with_suffix(extn0)

log0.info(f"loading: {if0}...")
data4 = list(srsly.read_jsonl(if0))
log0.info(f"loading: {if0}... DONE")

log0.info(f"{len(data4) = }")
```

```
I: loading: ../../data/d0000_word2vec-freq-ranks/word2vec_freq_ranks_raw.jsonl...
I: loading: ../../data/d0000_word2vec-freq-ranks/word2vec_freq_ranks_raw.jsonl... DONE
I: len(data4) = 3000000
```

4.3 prod: Frequency rank data dataframe

```
df4 = pd.DataFrame.from_records(data4)

df4["freq_idx"] = df4.freq_idx.apply(int)
df4 = df4.sort_values(by="freq_idx", ascending=False)
log0.info(f"{df4.shape = } [orig]")

df4["word"] = df4.word.str.lower()
df4["word"] = df4.word.apply(clean_str)
df4 = df4.drop_duplicates(subset="word", keep="first")
log0.info(f"{df4.shape = } [uniq]")

with warnings.catch_warnings():
    warnings.filterwarnings("ignore", message="This pattern is interpreted as a regular expression,
    cond4 = df4.word.str.contains(re0.pattern, regex=True, na=False, flags=re.IGNORECASE, case=Fals
```

```
df5 = df4[cond4]
df6 = df4[~cond4]

log0.info(f"{df5.shape = } [keep]")
log0.info(f"{df6.shape = } [drop]")
disp_df(df4.head(n=8))
disp_df(df4.tail(n=8))
```

```
I: df4.shape = (3000000, 2) [orig]
I: df4.shape = (2702147, 2) [uniq]
I: df5.shape = (700056, 2) [keep]
I: df6.shape = (2002091, 2) [drop]
```

	word	freq_idx
0	</s>	3000000
1	in	2999999
2	for	2999998
3	that	2999997
4	is	2999996
5	on	2999995
6	##	2999994
7	the	2999993

	word	freq_idx
2999991	shilpa_goenka	9
2999992	divider_pistons	8
2999993	thirsty_owl	7
2999994	righthander_kyle_drabek	6
2999996	bim_skala_bim	4
2999997	mezze_cafe	3
2999998	pulverizes_boulders	2
2999999	snowcapped_caucasus	1

4.4 prod: Expand patterns

```
n = 25
n = 12  # CAUTION
n = 10

df7 = df4.copy()
df7 = df5.copy()  # CAUTION

data7 = []
with Timer() as elapsed:
    for item0 in tqdm(data0):
        if item0.endswith("*"):
            pt7 = r"^" + item0.replace("*", r"[a-z\-]*") + r"$"
            cond7 = df7.word.str.contains(pt7, regex=True, na=False, flags=re.IGNORECASE, case=Fals
            words = df7[cond7].head(n=n).word.tolist()
            for word in words:
```

```
                if word not in data7:
                    data7.append(word)
        else:
            if item0 not in data7:
                data7.append(item0)

log0.info(hf.format_timespan(elapsed.elapsed))
log0.info(f"{len(data0) = }")
log0.info(f"{len(data7) = }")
```

```
100% 4618/4618 [02:43<00:00, 28.30it/s]
I: 2 minutes and 43.15 seconds
I: len(data0) = 4618
I: len(data7) = 10138
```

4.5 prod: Save

```
dir0 = "../../data/d0006_sources-expanded/"
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!"

bfn0 = dir0/"expanded"

xtn0 = ".jsonl"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
with open(ofn0, "w") as fh: pass
srsly.write_jsonl(ofn0, data7)

xtn0 = ".yaml"
ofn0 = bfn0.with_suffix(xtn0)
log0.info(f"saving: {ofn0}...")
with open(ofn0, "w") as fh: pass
srsly.write_yaml(ofn0, data7)

log0.info("DONE")
```

```
I: saving: ../../data/d0006_sources-expanded/expanded.jsonl...
I: saving: ../../data/d0006_sources-expanded/expanded.yaml...
I: DONE
```