Summary:

- 3 baselines, copy the value from last period, copy the value from the same period the year before, take the max between the last 2 periods
- Models on geo features alone can at best match the performance of the baseline
- Need to find a way to bring some improvement by adding conflict features!!

Gdelt events data processing

In [ ]:

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import os
import requests
import zipfile
from datetime import datetime, timedelta
from concurrent.futures import ThreadPoolExecutor
from bs4 import BeautifulSoup
import helper_functions.download_gdelt_events as download_gdelt
from helper_functions.download_gdelt_events import consolidate_files
from helper_functions.gdelt_data_mapping_optimized import load_gadm_data, pr
import multiprocessing
import nest_asyncio
nest_asyncio.apply()
from helper_functions.url_scraping_base import process_urls_in_chunks
import numpy as np
import glob
import re
import collections.abc

# ----------------------------------------------------------------------
# Download GDELT raw data
# ----------------------------------------------------------------------
start_date = datetime(2016, 1, 1)
end_date = datetime(2024, 2, 28)
save_directory = "../data/gdelt/events/1_raw"
output_file = "../data/gdelt/events/2_consolidated/combined_data.parquet"

# Paths
gadm_path = "../data/gadm/gadm_410_filtered_v2.gpkg"
fews_path = "../data/fews/fews_with_conflicts_admin2.parquet"
gdelt_path = "../data/gdelt/events/2_consolidated/combined_data.parquet"
output_dir = "../data/gdelt/events/3_mapped/"
final_output_path = os.path.join(output_dir, "gdelt_mapped.parquet")

# print("Downloading GDELT data...")
# download_gdelt.download_gdelt_data(start_date, end_date, save_directory, m

# print("Consolidating GDELT data...")
# consolidate_files(save_directory, output_file)
# 4M articles at this point
# recommended_cpus = max(1, multiprocessing.cpu_count() - 1)

# # # ---------------------------------------------------------------------
# # Load reference data
# # ---------------------------------------------------------------------
# fews_df = pd.read_parquet(fews_path)
# fews_df = fews_df[['ADMIN0', 'ADMIN1', 'ADMIN2', 'period', 'CS_score']]


# print("Loading GADM")
# gadm_gdf, fews_df = load_gadm_data(gadm_path, fews_df)

# # ---------------------------------------------------------------------
# # Load and process GDELT
# # ---------------------------------------------------------------------
# gdelt_df = pd.read_parquet(gdelt_path)
# gdelt_df = gdelt_df.dropna(subset=["ActionGeo_Lat", "ActionGeo_Long"])

# print("Processing GDELT data")
#
#  process_gdelt_data(gdelt_df, gadm_gdf, output_dir, num_cpus=recommended_c

# # ---------------------------------------------------------------------
# # Merge with FEWS
# # ---------------------------------------------------------------------
# print("Consolidating mapped GDELT data with FEWS data")

# df_final = consolidate_and_merge_fews(mapped_gdelt_dir=output_dir, fews_df
# # 4M articles at this point

# df_final.to_parquet(final_output_path, index=False)

# print("Number of records in final dataset:", len(df_final))
# print("\nFirst few rows of the final dataset:")
# df_final.head()
# 4M articles at this point
```

In [ ]:

```
df = pd.read_parquet(final_output_path)
# df = df[df['period'] == '202402']
# ---------------------------------------------------------------------
# Extract unique URLs
# ---------------------------------------------------------------------

def flatten_and_filter(source_col):
    urls = set()
    for item in source_col.dropna():
        if isinstance(item, (list, tuple, set, np.ndarray)):  # 👈 added np.
            urls.update(
                x for x in item
                if isinstance(x, str) and x.startswith("http")
            )
    return urls

unique_urls = sorted(flatten_and_filter(df["SOURCEURL"]))
```

```
# ---------------------------------------------------------------------
# Handle already processed chunks
# ---------------------------------------------------------------------
chunk_dir = "../data/gdelt/events/scraped_urls"
os.makedirs(chunk_dir, exist_ok=True)
parquet_files = glob.glob(os.path.join(chunk_dir, "chunk_*.parquet"))

chunk_numbers = (
    [int(re.search(r'chunk_(\d+)\.parquet', f).group(1)) for f in parquet_fi
    if parquet_files else []
)
highest_chunk = max(chunk_numbers) + 1 if chunk_numbers else 0
print(f"Highest chunk number: {highest_chunk}")

if parquet_files:
    processed = pd.concat([pd.read_parquet(f) for f in parquet_files], ignor
    processed_urls = set(processed['url'].dropna().values)
else:
    processed_urls = set()

# Remove already processed URLs
unique_urls = [u for u in unique_urls if u not in processed_urls]

print(f"Remaining URLs to process: {len(unique_urls)}")

# ---------------------------------------------------------------------
# Run URL scraping
# ---------------------------------------------------------------------
process_urls_in_chunks(
    urls=unique_urls,
    chunk_size=10000,
    concurrency=150,
    chunk_id=highest_chunk,
    timeout=4,
    max_retries=3,
    max_selenium_workers=4,
    fallback_mode="async_only",
    output_dir=chunk_dir
)
# 1.9M articles at this point
```

Clean the scraped URLS, remove duplicate articles, and articles not related to FS

In [1]:

```
import re
import os
import math
import logging
import multiprocessing as mp
from collections import Counter

from tqdm.notebook import tqdm
import pandas as pd
import glob
import spacy
```

```python
from helper_functions.topic_modelling.text_processing_parallel import prepro
from helper_functions.topic_modelling.flatten_articles import filter_article
from helper_functions.topic_modelling.deduplication import deduplicate_minha
from helper_functions.run_ner_parallel import run_ner_parallel, inject_count

# ----------------- Load scraped files -----------------
parquet_dir = "../data/gdelt/events/scraped_urls"
parquet_files = glob.glob(os.path.join(parquet_dir, "chunk_*.parquet"))
if not parquet_files:
    raise FileNotFoundError(f"No parquet files found in {parquet_dir}")

df_scraped = pd.concat([pd.read_parquet(f) for f in parquet_files], ignore_i
print("Total scraped articles:", len(df_scraped))
# ~1.9M articles at this point

# ----------------- Remove noisy events -----------------
mapped_path = "../data/gdelt/events/2_consolidated/combined_data.parquet"
if not os.path.exists(mapped_path):
    raise FileNotFoundError(f"Missing mapped GDELT file: {mapped_path}")

codes_mapping = pd.read_parquet(mapped_path)
codes_mapping = (
    codes_mapping
    .groupby("SOURCEURL", as_index=False)
    .agg({"EventCode": "max"})
)

df_scraped = pd.merge(df_scraped, codes_mapping, left_on='url', right_on='SO
df_scraped = df_scraped[~df_scraped["EventCode"].isin([10, 20, 30, 31, 32, 3
print("Remaining after removing 10/20/30/31/32/33/34/42/43/46:", len(df_scra
# ~920K here

# ----------------- Pre-processing pipeline -----------------
df_scraped['header'] = df_scraped['header'].fillna("").astype(str)
df_scraped['body'] = df_scraped['body'].fillna("").astype(str)
df_scraped['text'] = df_scraped['header'] + " " + df_scraped['body']

print("Applying LEAP4FNSSA lexicon filter (raw text)...")

df_scraped = filter_articles_by_lexicon(
    df_scraped,
    clean_text_col="text",
    lexicon_path="../data/LEAP4FNSSA_LEXICON_long.csv"
)
print("Lexicon filtering complete. Remaining:", df_scraped.shape)
# ~381k articles

# 2️⃣ Heavy preprocessing
df_scraped = preprocess_text_parallel(df_scraped, text_col='text')
print("Text preprocessing done.")

# 3️⃣ Truncate to 500 words
def truncate_to_500_words(text):
    words = str(text).split()
    return ' '.join(words[:500])
```

```python
df_scraped['clean_text'] = df_scraped['clean_text'].apply(truncate_to_500_wo

# 4️⃣ Deduplicate by near-duplicate text
df_scraped = deduplicate_minhash(df_scraped, text_col='clean_text', threshol
print("After deduplication:", df_scraped.shape)
# ~354k articles

# ------------------ NER Extraction ------------------
print("🔍 Running NER location extraction...")

# Inject country names for demonyms before NER
df_scraped["clean_text"] = df_scraped["clean_text"].apply(inject_countries_f

# Automatically disable multiprocessing in Jupyter (for stability)
n_process = 1 if "ipykernel" in mp.current_process().name.lower() else mp.cp

df_scraped = run_ner_parallel(df_scraped, text_col="clean_text", n_process=n
print("✅ NER extraction complete.")

# ------------------ Load FEWS countries (for both refinement & filtering) --
fews_path = "../data/fews/fews_with_conflicts_admin2.parquet"
fews_df = pd.read_parquet(fews_path)
fews_countries = [country.lower() for country in fews_df['ADMIN0'].unique()]

# ------------------ Refine main country using FEWS mentions -----------------
# Keep raw spaCy outputs for debugging if needed
df_scraped["NER_admin0_raw"] = df_scraped["NER_admin0"]
df_scraped["NER_admin1_raw"] = df_scraped["NER_admin1"]
df_scraped["NER_admin2_raw"] = df_scraped["NER_admin2"]

def pick_main_country_from_text(text: str) -> str:
    """
    Count mentions of each FEWS country in the article text (clean_text)
    and return the most frequently mentioned one (or None if no FEWS country
    """
    if not isinstance(text, str):
        return None

    txt = text.lower()
    counts = Counter()

    # naive substring counting; demonyms already injected as country names
    for country in fews_countries:
        c = txt.count(country)
        if c > 0:
            counts[country] += c

    if counts:
        return counts.most_common(1)[0][0]
    return None

def refine_ner_country(row):
    main = pick_main_country_from_text(row["clean_text"])
    if main:
        # override top-level country with the most-mentioned FEWS country
        row["NER_admin0"] = main
    return row

df_scraped = df_scraped.apply(refine_ner_country, axis=1)
print("✅ NER country refinement based on FEWS mentions complete.")

# ----------------- Filter by FEWS countries -----------------
df_scraped = df_scraped[
    df_scraped['NER_admin0'].isin(fews_countries) |
    df_scraped['NER_admin1'].isin(fews_countries) |
    df_scraped['NER_admin2'].isin(fews_countries)
]
print("Remaining after FEWS NER filter:", len(df_scraped))

# ----------------- Filter by crisis terms -----------------
crisis_terms = re.compile(
    r"(famine|hunger|malnutrition|food|nutrition|crop|harvest|yield|farmer|"
    r"agricultur|drought|flood|rainfall|storm|cyclone|heatwave|disaster|aid|
    r"refugee|displaced|idp|conflict|war|violence|attack|protest|unrest|"
    r"livelihood|poverty|shortage|inflation|market|commodity|price|"
    r"food security|early warning|ipc|wfp|unicef|ocha|ngo)",
    re.I
)

df_scraped = df_scraped[
    df_scraped["clean_text"].str.contains(crisis_terms, na=False)
]

print("After crisis-context filter:", len(df_scraped))

# ----------------- Save output -----------------
out_path = "../data/gdelt/events/scraped_urls/cleaned_filtered_urls.parquet"
df_scraped.to_parquet(out_path, index=False)
print(f"💾 Saved to {out_path}")
```

```
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports Ope
nSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. S
ee: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
Total scraped articles: 1922671
Remaining after removing 10/20/30/31/32/33/34/42/43/46: 921600
Applying LEAP4FNSSA lexicon filter (raw text)...
🔍 Filtering 921,600 articles using 386 lexicon keywords...
2025-11-23 12:10:51,885 - INFO - Starting parallel text preprocessing on 173
714 rows
2025-11-23 12:10:51,890 - INFO - Using 7 workers for parallel processing
2025-11-23 12:10:51,922 - INFO - Processing batch 1/18
✅ Retained 173,714 / 921,600 articles (18.85%) after lexicon filtering.
Lexicon filtering complete. Remaining: (173714, 6)
```

```
2025-11-23 12:11:16,018 - INFO - Processing batch 2/18
2025-11-23 12:11:54,267 - INFO - Processing batch 3/18
2025-11-23 12:12:13,474 - INFO - Processing batch 4/18
2025-11-23 12:12:37,006 - INFO - Processing batch 5/18
2025-11-23 12:13:03,639 - INFO - Processing batch 6/18
2025-11-23 12:13:15,376 - ERROR - Text processing error: string index out of range
2025-11-23 12:13:33,043 - INFO - Processing batch 7/18
2025-11-23 12:14:03,558 - INFO - Processing batch 8/18
2025-11-23 12:14:12,266 - ERROR - Text processing error: string index out of range
2025-11-23 12:14:32,935 - INFO - Processing batch 9/18
2025-11-23 12:15:11,810 - INFO - Processing batch 10/18
2025-11-23 12:15:17,937 - ERROR - Text processing error: string index out of range
2025-11-23 12:15:18,672 - ERROR - Text processing error: string index out of range
2025-11-23 12:15:19,783 - ERROR - Text processing error: string index out of range
2025-11-23 12:15:32,958 - INFO - Processing batch 11/18
2025-11-23 12:15:50,081 - ERROR - Text processing error: string index out of range
2025-11-23 12:15:50,092 - ERROR - Text processing error: string index out of range
2025-11-23 12:15:50,145 - ERROR - Text processing error: string index out of range
2025-11-23 12:16:01,413 - INFO - Processing batch 12/18
2025-11-23 12:17:06,523 - INFO - Processing batch 13/18
2025-11-23 12:17:32,649 - ERROR - Text processing error: string index out of range
2025-11-23 12:17:36,573 - INFO - Processing batch 14/18
2025-11-23 12:18:01,768 - INFO - Processing batch 15/18
2025-11-23 12:18:31,755 - INFO - Processing batch 16/18
2025-11-23 12:18:55,194 - INFO - Processing batch 17/18
2025-11-23 12:19:10,013 - ERROR - Text processing error: string index out of range
2025-11-23 12:19:23,007 - INFO - Processing batch 18/18
2025-11-23 12:19:34,039 - INFO - Text preprocessing completed successfully
Text preprocessing done.
Indexing for dedup: 0it [00:00, ?it/s]
Kept 156218 / 173714 unique articles (17496 removed).
After deduplication: (156218, 7)
🔍 Running NER location extraction...
🔍 Running NER extraction on 156218 texts with 8 processes...
```

```
Running NER in parallel (FEWS-aware):   0%|          | 0/156218 [00:00<?, ?i
t/s]/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-
packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports
OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.
3'. See: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports Ope
nSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. S
ee: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports Ope
nSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. S
ee: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] The rule-based le
mmatizer did not find POS annotation for one or more tokens. Check that your
pipeline includes components that assign token.pos, typically 'tagger'+'attr
ibute_ruler' or 'morphologizer'.
  warnings.warn(Warnings.W108)
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports Ope
nSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. S
ee: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] The rule-based le
mmatizer did not find POS annotation for one or more tokens. Check that your
pipeline includes components that assign token.pos, typically 'tagger'+'attr
ibute_ruler' or 'morphologizer'.
  warnings.warn(Warnings.W108)
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports Ope
nSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. S
ee: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports Ope
nSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. S
ee: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] The rule-based le
mmatizer did not find POS annotation for one or more tokens. Check that your
pipeline includes components that assign token.pos, typically 'tagger'+'attr
ibute_ruler' or 'morphologizer'.
  warnings.warn(Warnings.W108)
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports Ope
nSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. S
ee: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] The rule-based le
```

```
mmatizer did not find POS annotation for one or more tokens. Check that your
pipeline includes components that assign token.pos, typically 'tagger'+'attr
ibute_ruler' or 'morphologizer'.
  warnings.warn(Warnings.W108)
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] The rule-based le
mmatizer did not find POS annotation for one or more tokens. Check that your
pipeline includes components that assign token.pos, typically 'tagger'+'attr
ibute_ruler' or 'morphologizer'.
  warnings.warn(Warnings.W108)
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports Ope
nSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. S
ee: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
Running NER in parallel (FEWS-aware):   0%|          | 338/156218 [00:18<25:
53, 100.35it/s]/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/pyth
on3.9/site-packages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] Th
e rule-based lemmatizer did not find POS annotation for one or more tokens.
Check that your pipeline includes components that assign token.pos, typicall
y 'tagger'+'attribute_ruler' or 'morphologizer'.
  warnings.warn(Warnings.W108)
Running NER in parallel (FEWS-aware):   0%|          | 592/156218 [00:20<16:
59, 152.70it/s]/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/pyth
on3.9/site-packages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] Th
e rule-based lemmatizer did not find POS annotation for one or more tokens.
Check that your pipeline includes components that assign token.pos, typicall
y 'tagger'+'attribute_ruler' or 'morphologizer'.
  warnings.warn(Warnings.W108)
/Users/marco.bertetti/Desktop/git_repos/phd_nlp/venv/lib/python3.9/site-pack
ages/spacy/pipeline/lemmatizer.py:211: UserWarning: [W108] The rule-based le
mmatizer did not find POS annotation for one or more tokens. Check that your
pipeline includes components that assign token.pos, typically 'tagger'+'attr
ibute_ruler' or 'morphologizer'.
  warnings.warn(Warnings.W108)
Running NER in parallel (FEWS-aware): 100%|██████████| 156218/156218 [40:08<
00:00, 64.87it/s]
```

```
✅ NER extraction complete (FEWS-aware).
✅ NER extraction complete.
✅ NER country refinement based on FEWS mentions complete.
Remaining after FEWS NER filter: 135903
After crisis-context filter: 131727
💾 Saved to ../data/gdelt/events/scraped_urls/cleaned_filtered_urls.parquet
```

In [2]:

```python
df_scraped[df_scraped['url'] == 'https://www.thenews.com.pk/print/666155-30-
```

Out[2]:

	url	header	body	
70529	https://www.thenews.com.pk/print/666155-30-kil...	30 killed in eastern Burkina Faso attack	OUAGADOUGOU: Gunmen killed around 30 people at...	https://www.t