UDC 004.9:004.4 DOI: https://doi.org/10.64076/iedc250806.18

Software pipeline development for OSINT-based geocoding of missile and drone strikes in Ukraine

Maksym Ilin
National Technical University of Ukraine "Igor Sikorsky Kyiv polytechnic institute", Kyiv
https://orcid.org/0009-0001-0803-3726
Liubov Oleshchenko
National Technical University of Ukraine "Igor Sikorsky Kyiv polytechnic institute", Kyiv
https://orcid.org/0000-0001-9908-7422

Abstract. *A local offline-first software pipeline was developed to extract and geocode missile/drone strike data in Ukraine from open-source reports. It features multilingual event detection, morphology-aware toponym extraction via lemmatization, and offline geocoding using a local GeoNames database, producing reproducible impacts.csv datasets.*
Keywords: *software pipeline development, OSINT, geocoding, missile/drone strikes.*

There is a critical need for an automated, reliable, and offline-capable software pipeline to extract and geocode missile and drone strike events in Ukraine from multilingual and unstructured open-source intelligence (OSINT) reports, with the aim of generating a reproducible and structured dataset to support situational awareness, analysis, and humanitarian response. Telegram's desktop client supports chat exports for offline analysis, enabling structured JSON processing without scraping. Established projects show the value of systematically coded open-source reports: ACLED emphasises analyst-curated geo/time precision tags [1], while GDELT automates multilingual event monitoring [2]. OSINT practice routinely augments automation with map-based leads, e.g., Bellingcat's OSM search tooling [3].

A persistent gap for Ukraine is handling inflected UA/RU toponyms and avoiding API rate limits; OSM's Nominatim policy explicitly discourages bulk queries and encourages local data use [4]. We, therefore, propose an export-only, offline, multilingual micro-pipeline from text to impacts.csv.

Inputs are Telegram Desktop JSON exports and news archives (CSV/JSON). We detect impact events using curated UA/RU trigger terms and filter negations. Timestamps are normalized to UTC; date-only items are tagged with lower time precision. Reports remain unverified OSINT; this stage does not deduplicate across sources.

Stanza [5] provides UA/RU NER and lemmatization, improving recall on inflected toponyms (Table 1).

spaCy's Ukrainian model is a viable alternative but is secondary here [6].

Table 1. Stanza toponym extraction (snippet)

```
import stanza
stanza.download("uk", processors="tokenize,ner,lemma", verbose=False)
nlp = stanza.Pipeline("uk", processors="tokenize,ner,lemma", tokenize_no_ssplit=True)

def extract_places(text: str) -> list[str]:
 doc = nlp(text)
 places = []
 for ent in doc.entities:
 if ent.type == "LOC":
 lemmas = []
 for token in ent.tokens:
 for word in token.words:
 lemmas.append(word.lemma or token.text)
 places.append(" ".join(lemmas))
 return places
```

Local GeoNames UA + alternateNames enables API-free geocoding with rich variants/transliterations [7] (Table 2).

Table 2. Offline GeoNames lookup (snippet)

```
import pandas as pd
from math import inf

gdf = pd.read_parquet("geonames_UA.parquet") # cols: name, alternatenames, lat, lon, pop, fcode
gdf["name_lc"] = gdf["name"].str.casefold()
gdf["alts_lc"] = gdf["alternatenames"].fillna("").str.casefold()

def edit_distance(a,b):
 dp=[[i+j if i*j==0 else 0 for j in range(len(b)+1)] for i in range(len(a)+1)]
 for i in range(1,len(a)+1):
 for j in range(1,len(b)+1):
 dp[i][j]=min(dp[i-1][j]+1,dp[i][j-1]+1,dp[i-1][j-1]+(a[i-1]!=b[j-1]))
 return dp[-1][-1]

def resolve_toponym(q: str):
 ql = q.casefold()
 exact = gdf[(gdf["name_lc"]==ql) | (gdf["alts_lc"].str.contains(ql))]
 if not exact.empty: # prefer highest pop
 return exact.sort_values("pop", ascending=False).iloc[0]
 best, row = inf, None
 for _, r in gdf.iterrows():
 d = edit_distance(r["name_lc"], ql)
 if d < best:
 best, row = d, r
 return row if best < 3 else None
```

Case-insensitive exact match first; then controlled fuzzy match (edit distance/trigrams) within UA entries; tie-breaks by population/feature code.

For each detected event, a confidence level was assigned to reflect data reliability: High (exact match), Medium (alternate interpretation or tie-break), and Low (fuzzy match). Precision tags followed the ACLED standard to indicate the accuracy of time and location data.

The results were saved to the impacts.csv file, which includes six main fields: timestamp_utc, lat, lon, event_type, source, and text. Additionally, optional fields location_precision and confidence were used to enhance data interpretability (Table 3).

Table 3. impacts.csv schema
columns = ["timestamp_utc", "lat", "lon", "event_type", "source", "text"]

The proposed software system ingests only offline exports (no scraping) and outputs a geocoded impact dataset. NLP steps handle multilingual text and morphological normalisation, and gazetteer resolution is done locally with confidence and precision metadata (Fig. 1).

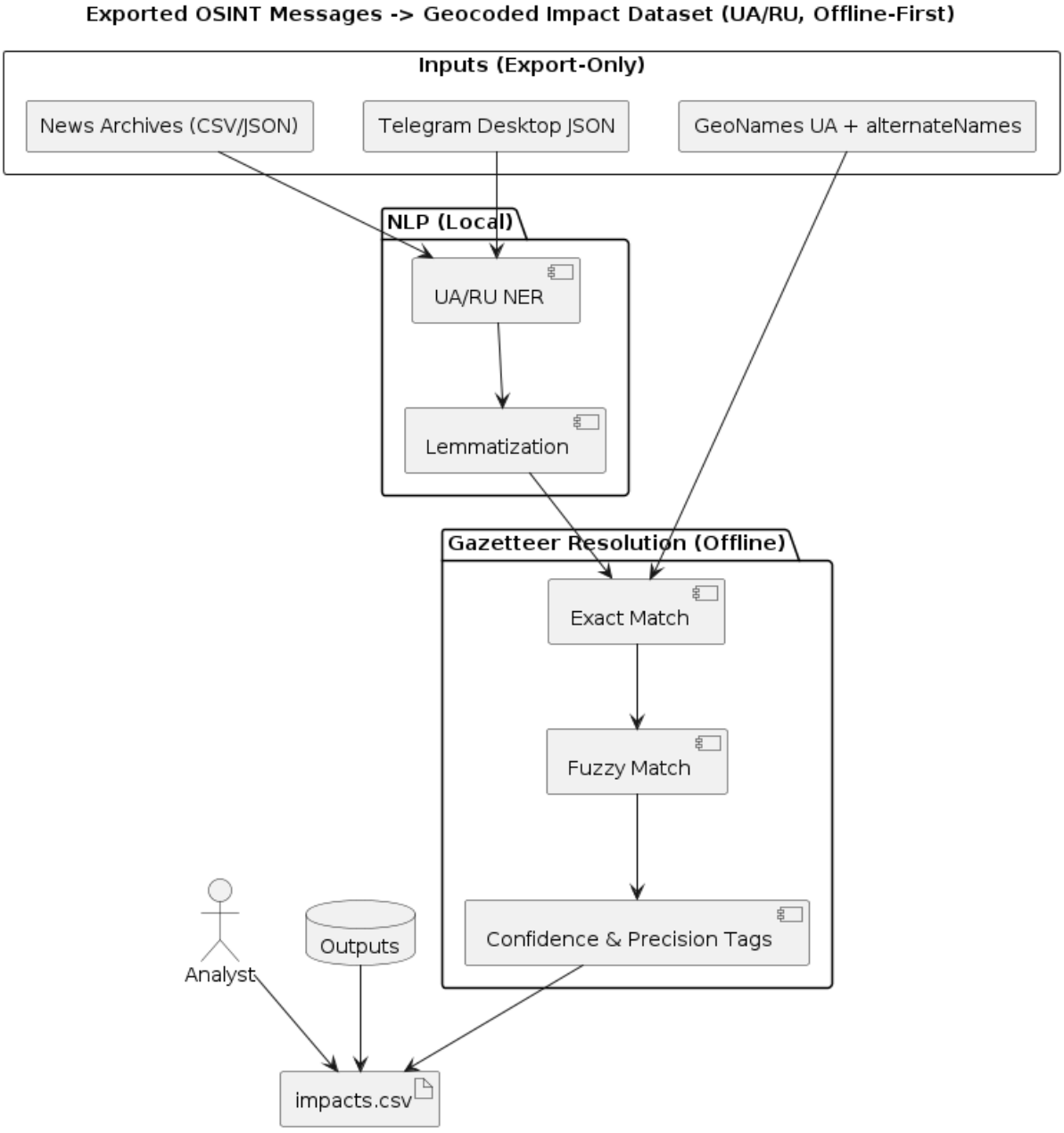

Fig. 1. Proposed exported OSINT message processing pipeline

Export-only inputs, morphology-aware toponyms (NER+lemma), and offline multilingual geocoding (GeoNames alternates) yield higher recall and reproducibility without API limits. Transparent uncertainty via confidence flags and ACLED-style precision supports downstream filtering and fusion. The pipeline is modular, auditable, and portable to other regions/languages.

Future work will target three tightly scoped upgrades. First, event detection and temporal normalisation can shift from keyword rules to a compact local classifier that models context, negation, and speculation, paired with rule-plus-ML time parsing to convert vague expressions into UTC timestamps with explicit confidence and standardised time-precision codes.

Second, toponym resolution and gazetteer quality can be improved by enforcing administrative constraints and name co-occurrence, adding lightweight coreference to maintain intra-source consistency, and periodically enriching the offline gazetteer with curated alternates and historical forms reconciled via stable identifiers to reduce fuzzy matches and support longitudinal analyses.

Third, cross-source deduplication and evaluation can be formalised by clustering reports in joint space–time windows to merge near-duplicates while retaining provenance, and by establishing a small stratified gold standard to report precision/recall by region, language, and settlement size, creating measurable baselines for iterative improvement.

References

1. ACLED (2021). ACLED codebook (Version 1). URL: https://acleddata.com/acled-codebook.

2. Voukelatou V. (2020, July 28). The GDELT Project: Uniquely massive open dataset for global society insights. SoBigData Blog. URL: https://www.sobigdata.eu/blog/gdelt-unique-massive-and-open-dataset.

3. Logan Williams. Bellingcat. (2023, May 8). Finding geolocation leads with Bellingcat's OSM search tool. URL: https://www.bellingcat.com/resources/how-tos/2023/05/08/finding-geolocation-leads-with-bellingcats-openstreetmap-search-tool.

4. OpenStreetMap Foundation. (2023). Nominatim usage policy. URL: https://operations.osmfoundation.org/policies/nominatim.

5. Stanford NLP Group (2021). Stanza: Named entity recognition models – performance tables (v1.4.1). URL: https://stanfordnlp.github.io/stanza/ner_models.html.

6. Explosion AI (2023). spaCy Ukrainian model (uk_core_news_md) [Software]. URL: https://spacy.io/models/uk.

7. GeoNames (2023). GeoNames geographical database: Data dump (UA) [Data set]. URL: https://download.geonames.org/export/dump.