A Geoparsing Pipeline for Multilingual Social Media Posts from Ukraine

Maxim Mironov[1,†], Alexander Marquard[1,†], Daniel Racek[1,*], Christian Heumann[1], Paul W. Thurner[2] and Matthias Aßenmacher[1,3,*]

[1]Department of Statistics, LMU Munich, Munich, Germany
[2]Geschwister Scholl Institute of Political Science (GSI), LMU Munich, Munich, Germany
[3]Munich Center for Machine Learning (MCML), Munich, Germany

Abstract

The dynamics of contemporary social media communication, particularly on platforms like X (formerly Twitter), have significantly evolved, and this data is frequently used for scientific research. However, due to X's API changes in 2019, a tweet's precise geolocation is no longer present in the data, thus preventing a geographical assessment of tweets. This project aims to extract location mentions from tweets' texts and to map them to Ukraine's administrative regions. We have developed a specialized pipeline for geoparsing with specific prebuilt components for the Ukrainian, Russian, and English languages. The main advantage of our pipeline's architecture is the interchangeability of all components, allowing for the integration of custom-developed solutions. Initial tests on our hand-labeled Ukrainian dataset show promising results in accurately identifying and mapping location mentions despite various challenges, such as declension and the presence of multiple languages in a single tweet. Additional experiments using publicly available benchmark data further indicate promising performance when transferring our pipeline to other geographical regions. Both our geoparsing pipeline and its online documentation have been made publicly available.

Keywords

Location Reference Recognition, Geoparsing, Natural Language Processing, Named Entity Recognition

1. Introduction

Geotagged social media posts constitute a valuable resource for researchers across numerous fields, including hazard management [1], public health [2] and politics [3]. However, in 2019, X (formerly Twitter), one of the most important platforms for such research [4, 5], removed the option for precise geotagging [6]. This has prompted researchers to increasingly develop methods for extracting geolocation data from textual information of social media posts and mapping these to specific coordinates [7]. This process, in the literature commonly referred to as geoparsing [7], comes with various difficulties. Social media posts are often multilingual, sometimes even written in multiple languages at once, and frequently contain misspellings and informal language. Moreover, determining whether a post refers to a specific location depends on its broader context. Geoparsing consists of two primary steps, location mention recognition (also known as location reference recognition or toponym recognition), which detects mentioned locations in text. Second, geocoding (also known as toponym resolution), which identifies and assigns geographic coordinates to these mentioned locations [8]. While numerous studies have focused on location mention recognition [9], full geoparsing pipelines are scarce, often lack transparency, and are limited to the most commonly spoken languages such as English, making them unsuitable for many research projects.

Contributions. In this work, we present a fully transparent geoparsing pipeline designed for tweets

GeoExT 2025: Third International Workshop on Geographic Information Extraction from Texts at ECIR 2025, April 10, 2025, Lucca, Italy

*Corresponding author.
†These authors contributed equally.
maxim.mironov@campus.lmu.de (M. Mironov); a.marquard@campus.lmu.de (A. Marquard); daniel.racek@stat.uni-muenchen.de (D. Racek); chris@stat.uni-muenchen.de (C. Heumann); paul.thurner@gsi.uni-muenchen.de (P. W. Thurner); matthias@stat.uni-muenchen.de (M. Aßenmacher)

from Ukraine, written in the three most prevalent languages, Ukrainian, Russian, and English. Our geoparser **TBGAT** (**t**ext-**b**ased **g**eographical **a**ssignment of **t**weets) matches each tweet to the geographic coordinates of the mentioned locations. We compare our method to a spaCy-based geoparser, showcase its superior performance, and analyze the locations of tweets before and during the Russian invasion of Ukraine, used for analyzing language use in [10]. Although designed for tweets from Ukraine, the pipeline is highly adaptable and can be used to match any social media posts to Ukrainian locations. Furthermore, extensions to other regions and languages are easily possible using our publically available implementation[1] , as we also showcase on the IDRISI-RE benchmark dataset [11].

2. Related Work

To date, the majority of studies have focused on location mention recognition (LMR). As noted by [9], these approaches can be broadly categorized into rule-based methods, gazetteer matching, statistical learning, or hybrids of these techniques. Rule-based approaches rely on predefined rules, such as regular expressions, to identify recurring patterns that denote location mentions. However, defining a comprehensive and robust set of rules remains challenging. Gazetteer-based methods match n-grams from the text with entries in location dictionaries along with additional heuristics for filtering and disambiguation. The main challenges for this approach include collecting a complete set of locations and addressing variations in names and context-specific ambiguities. Statistical models are trained on annotated text corpora, learning to identify and extract location information from unlabeled previously unseen texts. A large strand of literature employs Named Entity Recognition (NER), which classifies portions of text as specific types of entities, including location entities. In recent years, with the emergence of large language models (LLMs) such as BERT [12] and its successors, the focus has shifted towards improving these deep learning-based NER models for LMR. Nonetheless, these models face obstacles such as limited availability of annotated training data or handling social media's frequent misspellings and informal language. Hence, hybrid approaches, which combine any of the aforementioned techniques, have been designed to overcome some of these issues. For an extensive review and comparison of LMR methods, we refer to the survey by [9].

Geoparsing, which combines location mention recognition with geocoding, plays a crucial role across many disciplines. Applications include, among others, disaster response [13, 14], disease surveillance [15], traffic control [16], crime management [17], geographic information retrieval [18]. However, the number of freely available geoparsing tools is limited and most are not actively maintained. The complexity of geoparsing arises from the unique characteristics of each use case, which vary by the type of text (e.g., social media vs. news articles), language(s) present, and the geographic area to be considered. Each language requires a customized approach to LMR, affecting all methods similarly. Moreover, handling misspellings typically requires use-case-specific solutions. To achieve optimal performance, geoparsing pipelines must also be set to a certain geographic area and level of granularity for the matching process (e.g. administrative zone vs. street level). All of these factors contribute significantly to the complexity of designing and implementing an effective (open-source) geoparser.

3. Methodology

For the development of our geoparsing pipeline, we considered four key aspects: efficiency, accuracy, transparency, and customizability. The simplicity of our pipeline facilitates easy customization and extension. Moreover, all modules within our pipeline are interchangeable, allowing for the integration of custom-developed solutions. In short, the structure of the pipeline can be described as follows:

1. Preprocessing the texts to ensure they comply with the requirements of the subsequent modules. This involves text normalization, clearing of whitespace, and other necessary adjustments.

[1] https://github.com/Maxim-M-D/tbgat/

2. Detecting the language(s) used in the tweet. In cases where multiple languages are present, the text is split into appropriate parts for further processing.
3. Next, we perform location mention recognition (LMR) using:
    - Gazetteers, employing the Aho-Corasick algorithm [19] for efficient pattern matching.
    - Transformer models, fine-tuned on the NER task.
4. We then map the locations identified in step three to coordinates obtained using OSM (geocoding).
5. Lastly, we conduct various post-processing tasks such as a further mapping of the obtained coordinates to first-level administrative regions (Ukrainian Oblasts), checking for special cases in the found locations, and output formatting.

3.1. Dataset & Labelling

We are using a dataset of tweets from [10], who have studied language use on X (formerly Twitter) in Ukraine before and during the Russian invasion. We draw on the ~ 2.3M tweets in the three most common languages (English, Ukrainian, and Russian) and match these to the geographic coordinates of the mentioned locations using our pipeline. To evaluate quality and performance, we draw a (stratified) random sample of 3000 tweets in total, consisting of 1000 tweets in each language (English, Ukrainian, and Russian). This sample was then manually labeled by a native speaker, who extracted the mentioned location[2] exactly as it appeared in the tweet. Additional geo-information in the form of latitude and longitude was added, based on the coordinates available in OpenStreetMap (OSM)[3], a collaborative project that provides freely accessible geographic data, which has been previously employed for similar applications [20]. These labeled location coordinates are then compared to the coordinates assigned by our geoparsing pipeline.

3.2. Pipeline Components

Having discussed the data labeling process we now turn our attention to the internal structure of the pipeline, namely the modules, which are specifically designed to handle geoparsing in the context of processing tweets from Ukraine in English, Russian, and Ukrainian respectively. However, we encourage adaptations for a broader range of geoparsing tasks. On a high level, the pipeline components can be separated into three distinct modules: processing-, extraction- and mapping module (cf. Fig. 1).

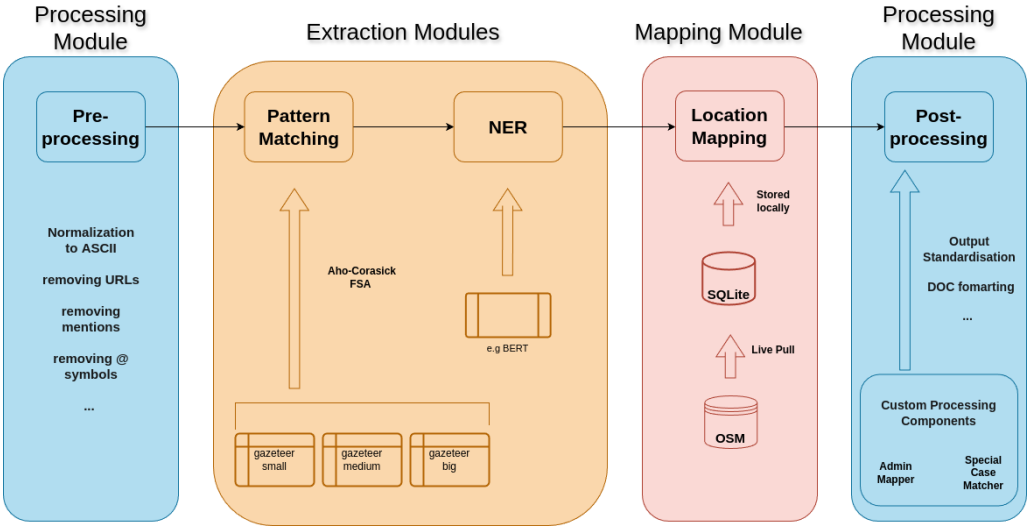

Figure 1: Overview of the three components of the geoparsing pipeline.

[2]Locations only in Ukraine were labeled
[3]https://www.openstreetmap.org/, for more details see Appendix B

The processing module consists of two individual components: The pre-processing module, which deals with normalizing the data while detecting the language in which the tweet is written, and the ”Special case matcher module” which in the context of tweets from Ukraine deals with the extraction and mapping of the occupied territories and common misspellings that were otherwise not possible to map. After normalizing the data and assigning a language to the tweet in the pre-processing module, the geographical information present in the text is extracted via the extraction modules. First, pattern matching is applied via gazetteers after which a NER model extracts further geographical details. This information is then mapped by the ”mapping module” using the (locally stored) openly available OSM data. The subsequent ”special case matcher module” helps to map edge cases, for which no OSM data is available, enforcing a final quality check.

3.3. Processing Module

Normalization. While the use of cursive writing and emojis on social media allows for creative expression, these features pose challenges for LMR, complicating not only pattern matching with gazetteers and NER but also earlier steps such as language identification.

In order to combat this, we start by normalizing tweets to use standard ASCII characters. For example **kharkiv** might be encoded as ”1D48C 1D489 1D482 1D493 1D48C 1D48A 1D497” in hexadecimal representation, which highlights the usage of non-standard ASCII characters. For reference, the standard hex representation of **kharkiv** using standard ASCII characters is ”6B 68 61 72 6B 69 76”. At the same time, we would like to keep Cyrillic letters as in Russian or Ukrainian tweets. For this task, we utilize the Python library *unicodedata*[4] which can be used to normalize strings according to the normal form KD (NFKD - normal form canonical decomposition), i.e. it replaces all compatibility characters (like U+00C7, the Latin capital letter C with cedilla) with their equivalents (here U+0043 - the Latin capital letter C). Further, we apply normalizations in the form of removing URLs, removing mentions and @-Symbols, removing hashtag symbols, removing emojis and finally removing extra whitespace.

Language Detection. The next step in the pre-processing module is language identification. While most texts in many applications are monolingual, the scenario shifts significantly for microblogs, where the phenomenon of code-switching, resulting in code-mixed texts (texts composed of multiple languages) is prevalent. Research on language identification has traditionally focused on monolingual texts [see e.g. 21, 22, 23], with comparatively minimal attention given to code-mixed texts. One challenge specific to language identification in microblogs, such as tweets, is their shortness, which complicates the effectiveness of many language identification tools including Google’s CLD2 or CLD3, FastText, and langid [24]. To address the issue of potential multilingualism in a tweet we utilize the Ligua Python package[5], which provides an innovative approach and not only offers superior accuracy but also enhanced (computational) performance over conventional methods. Lingua employs a probabilistic n-gram model utilizing the character distribution from a training corpus, extending the typical tri-gram model to include n-grams ranging from 1 to 5 in size. This extension allows for more accurate language predictions, particularly in shorter texts where fewer n-grams are present, and thus, the probability estimates from these n-grams are less reliable. Lingua additionally incorporates a rule-based engine that complements its statistical model. This engine initially identifies the alphabet of the input text and searches for characters unique to specific languages. If a single language can be conclusively identified through this method, the statistical model is not required. The rule-based engine also serves to eliminate languages that do not meet the criteria of the input text before the probabilistic n-gram model is considered. This not only saves memory but also improves runtime performance by reducing the number of language models loaded.

[4] https://docs.python.org/3.11/library/unicodedata.html
[5] https://github.com/pemistahl/lingua-py

3.4. Extraction Modules

Pattern Matching: Aho-Corasick FSA. For pattern matching, we employ the Aho-Corasick FSA [19], a specialized algorithm designed to efficiently handle multiple pattern searches simultaneously. The Aho-Corasick algorithm constructs a finite state automaton from a set of strings, effectively creating a trie (pronounced tree, but originates from *retrieval*) structure with additional failure links. These failure links connect each node to the next node that represents the longest possible suffix of the string corresponding to the current node. This structure allows the algorithm to transition between trie nodes without requiring a restart from the root, thus avoiding unnecessary reprocessing of the input text [19], cf. Appendix A.

Pattern Matching: Gazetteers. For our gazetteers, we compile city names from various sources to enhance the accuracy and comprehensiveness of our location-mention recognition system. This compilation involves a manual collection of city names from Wikipedia[6] which provides approximately 230 city names across three languages: English, Ukrainian, and Russian. Additionally, we utilize the Humanitarian Data Exchange Project[7] to obtain a more extensive list, encompassing roughly 100,000 populated places in Ukraine, also available in English, Ukrainian, and Russian. These diverse sources are integrated to create three distinct gazetteers, each varying in granularity. The smallest dataset focuses solely on locations within the first and second administrative regions, offering a more targeted data set. The medium-sized data set expands this scope to include locations from the first, second, and third administrative regions, providing a broader, yet still manageable, collection of locations. The largest dataset encompasses locations from all four administrative regions, offering the most comprehensive coverage. This tiered approach allows for flexible application of the gazetteers based on the specific needs of the task, whether it requires detailed granularity or extensive coverage.

Named Entity Recognition. Related work [25] effectively illustrates the limitations inherent in models that rely solely on gazetteers for geographical coverage. These models can be overly restrictive and may also lead to mismatches due to their inability to contextualize the data they process. For instance, during preliminary analysis, we identified villages in Ukraine named ”Lazy” and ”Smile”. However, the context-unaware nature of gazetteers led to the erroneous recognition of the common words ”lazy” and ”smile” as location names when they appear in standard conversations. To tackle context-unawareness we additionally employ NER. We observed that using individual NER models trained on specific languages performed better than a single multilingual model. Specifically for the ”quality” [8] version of pipeline we utilize the following BERT-based models from Huggingface [26], each fine-tuned for NER tasks tailored to different languages and data sets:

- For English, we employ a fine-tuned BERT on the task of NER on X data [27]. This model was trained on a corpus of 154 million tweets, making it highly adept at recognizing and classifying named entities within the informal and often abbreviated language commonly found on X.
- For Russian tweets, we utilize a fine-tuned BERT on the AmazonScience MASSIVE data set [28, 29].
- Ukrainian tweets were processed using another BERT-based model [30] fine-tuned on the Slavic-NER dataset[9].

By employing these specialized, fine-tuned models, our pipeline is well-equipped to handle the intricacies and linguistic variations present in tweets across these three languages.

[6] https://en.wikipedia.org/wiki/List_of_cities_in_Ukraine
[7] https://data.humdata.org/dataset/ukraine-populated-places
[8] See Results section
[9] https://github.com/SlavicNLP/SlavicNER

3.5. Mapping Module

After a location mention has been extracted via NER and the gazetteers, it is passed on to the mapping module. For this purpose, we utilize OSM. The integration of OSM in our pipeline allows us to leverage its comprehensive and up-to-date geographic database to associate each identified location with its corresponding administrative region. To implement this, once locations are identified and extracted from the tweets, each location name is queried against the OSM database to retrieve its geographic coordinates. Employing OSM for this task offers several advantages, including access to a global scope of geographic information and the ability to receive updates and corrections from a vast community of users. This ensures that our location mapping remains accurate and reflects current administrative boundaries, thereby improving the reliability of the subsequent analysis or application that relies on this geocoded data.

3.6. Post-Processing Module

The ”Post-Processing Module” is the last component in the pipeline and is used for output-related tasks such as output formatting and in our special use case the admin-level mapper and the special case matcher.

Admin-level Mapper. The retrieved geographical coordinates are then used to verify whether a given location lies within an administrative region by comparing it with the geographical properties of the administrative regions, which we obtained from the Humanitarian Data Exchange project.[10]

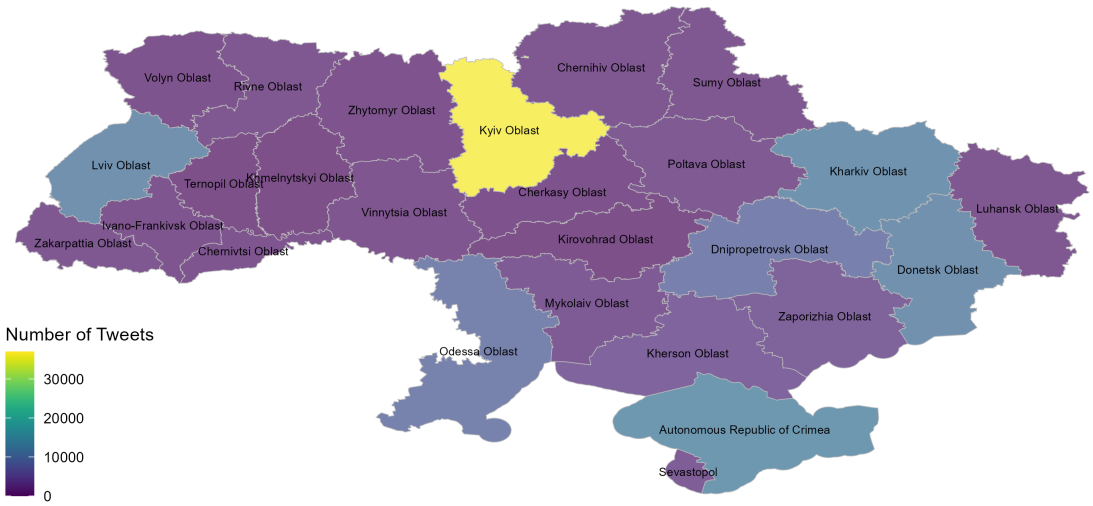

Figure 2: Map of Ukraine: Geographical distribution of the detected location mentions across the country.

Special case matcher As a last step before output formatting, we use the special case processing module that addresses some of the issues specific to our task, such as the presence of occupied territories and common but extreme misspellings of location names. Such problems present substantial challenges in accurately assigning geographic data to the correct administrative regions, particularly because our methodology heavily relies on the ability to query geographical information from OSM successfully. To address these special cases, we implement a dictionary-based approach involving predefined dictionaries that contain the properties of locations known to be problematic. Each entry in these dictionaries includes the correct mapping for a location, accounting for its unique circumstances.

[10] https://data.humdata.org/dataset/geoboundaries-admin-boundaries-for-ukraine

Output Formatting. The last step of the pipeline is the formatting of the output. As of the time of writing this paper the pipeline's current output provided by the Post-processing layer, can be characterized as follows: Similar to spacy we return a "DOC object" for each individual row of the data. The DOC object can be thought of a list that contains all information found by the pipeline. This includes the found location, position in the sentence, geographical information, and the obtained administrative level of the found location among other information[11].

4. Experimental Results

4.1. Ukraine Benchmark

We evaluate two variants of our pipeline: a performance version, which in the extraction module only uses the pattern matching component to speed up the geoparsing process, and a quality version, which in addition utilizes the NER component as described in Section 3.4. We compare our results with the spaCy-based python package geoparser[12] [31] on our labeled Ukrainian dataset described in Section 3.1, using both Accuracy and F1 score across all three languages. We additionally report average GPU runtimes across three runs on a common consumer-level GPU (NVIDIA RTX 3070) as well as CPU runtimes on an I5-1345U Intel CPU.

As shown in Table 1, TBGAT performs well in both predictive accuracy and computational efficiency. Compared to geoparser, it improves overall accuracy by up to 2.1 percentage points (+4.1%) and increases the F1 score by 0.19 (+38.8%). Performance increases can be observed across the board for both types of pipelines. GPU runtime is reduced by a factor of 3.6, reaching up to 11.4 in the performance-optimized version. This reduction in runtime is essential for making it feasible to process and match millions of tweets to locations. Additionally, we observe performance differences between the languages, with the performance-optimized pipeline, which excludes NER, slightly outperforming the quality-focused version for Russian tweets.

Table 1
Ukraine Benchmark Performance Results

	Runtime (sec)		Accuracy				F1-Score			
Model	GPU	CPU	EN	UKR	RU	Overall	EN	UKR	RU	Overall
TBGAT-Quality	37	1040	**0.923**	**0.882**	0.946	**0.916**	**0.816**	**0.498**	0.545	**0.680**
TBGAT-Performance	**12**	**69**	0.872	0.866	**0.949**	0.895	0.640	0.367	**0.581**	0.545
geoparser	136.7	1784	0.881	0.844	0.922	0.880	0.720	0.098	0.091	0.490

4.2. IDRISI-RE Benchmark

To validate our approach and compare its performance against a well-known benchmark dataset, we use the publicly available IDRISI-RE dataset [11], selecting all English-based sub-datasets located in the US. As shown in Table 2, our pipeline, despite being originally designed for tweets from Ukraine only, can easily be adapted to other countries and regions by exchanging individual components, as the performance results are competitive.[13] Since our framework allows for an easy exchange of components, refinements in the pipeline would further improve this performance.

[11]For more details please see the GitHub repository.

[12]Geoparser is one of the very few geoparsing libraries based on state-of-the-art NER from spaCy, which is also actively maintained. In contrast, most other geoparsing libraries are difficult to set up, often due to complex installation processes or other technical challenges.

[13]In the extraction module, for simplicity, we remove the pattern matching and only rely on a roBERTa-based NER model [32]. We additionally set our mapping module, i.e. OSM, to the US.

Table 2
IDRISE-RE Performance Results

Dataset	TBGAT (custom)		geoparser	
	F1	Acc	F1	Acc
California Wildfires 2018	0.723	0.566	**0.754**	**0.606**
Hurricane Dorian 2019	**0.522**	**0.352**	0.479	0.315
Hurricane Florence 2018	**0.540**	**0.370**	0.419	0.265
Hurricane Harvey 2017	**0.790**	**0.790**	0.727	0.572
Hurricane Irma 2017	**0.639**	**0.469**	0.579	0.407
Hurricane Maria 2017	0.171	0.09	**0.543**	**0.372**
Hurricane Matthew 2016	**0.17**	**0.09**	0.092	0.048
Maryland Floods 2018	0.792	0.493	**0.820**	**0.695**
Midwestern US Floods 2019	0.881	0.788	**0.917**	**0.848**

4.3. Ukranian Tweet Analysis

In addition to the benchmarks, we also applied our pipeline to all ~ 2.3M tweets, for which we present our findings below. We plot the geographical distribution of tweets across all first-level administrative zones in Figure 2, revealing that most tweets mention Kyiv Oblast, followed by Crimea and Kharkiv.

In Figure 3 we visualize the number of tweets over time for the five administrative zones with the most location mentions. As evident from the graphs, there is a clear spike with the start of the Russian invasion. Generally, the spikes for the different locations seem to align with the course of the war. For example, Kyiv Oblast was mainly targeted in the early stages of the invasion. By the start of April, most of the Russian troops were successfully forced out, which is also noticeable in the decrease in tweets. Another example is Kharkiv Oblast. In September, the Ukrainian troops launched a major counteroffensive, in which they successfully reclaimed multiple cities in the Oblast. In our data, this offensive coincides with a major increase in tweets mentioning Kharkiv Oblast during that time.

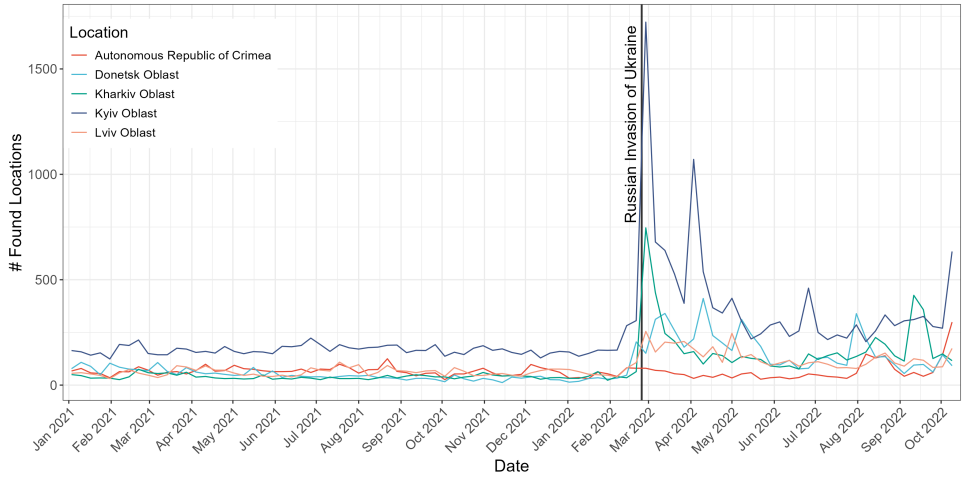

Figure 3: Development of the weekly number of detected location mentions over time. The black vertical line indicates the start of the Russian invasion.

5. Discussion and Limitations

Our TBGAT pipeline successfully matches multilingual tweets from Ukraine to their mentioned locations. Notably, its use extends beyond tweets, as it is similarly capable to match any other social media post to Ukrainian locations. While the flexibility of our pipeline offers many advantages over other available packages, it has still several limitations. One of the biggest limitations concerns the complexity of Russian and Ukrainian grammar. Russian and Ukrainian are both characterized by strong declension[14].

[14]'declension': the inflection of nouns, pronouns, or adjectives for case, number, and gender, for an example see Appendix

This often hinders the ability to map the identified locations in text with the OSM layer, as the declined locations cannot be found in the OSM database. A possible solution to this would for example be the introduction of a ”language normalization” module, specific to each language.
Another problem is that the pipeline cannot account for heavy misspellings, which regularly take place in social media posts. Furthermore, the ongoing renaming of Ukrainian cities poses a challenge. While the renaming can be tackled on an administrative level via OSM, intra-personal communication cannot be strictly mandated, and some inhabitants still refer to cities by their old name as we observe in our data. Due to this, the matching of a mentioned location to coordinates is not always possible, as the old city name may simply not exist anymore in any of the geographical databases. An expansion of our ”special case matcher” in the post-processing module can potentially offset this issue, however, this requires regular updating in order to guarantee correctness.
Finally, we want to make researchers aware that X (formely Twitter) recently has, similar to many other social platforms, severely restricted (research) access to their API. Additionally, data sharing is also usually either restricted or entirely forbidden according to social media platforms' legal terms [33].

6. Outlook

We have identified several potential goals for future work. First, we would like to implement a custom module that normalizes Russian and Ukrainian tweets in order to tackle declension. Second, fine-tuning a NER model for location recognition on a more granular level, to e.g. map specific locations such as Maidan square, is another avenue for future work, as it is currently not possible to identify these correctly. Finally, a deeper analysis of the tweets and their corresponding locations in Ukraine with respect to the invasion should prove promising for political scientists to better understand all facets of the war.

Acknowledgments

Matthias Aßenmacher was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure – NFDI 27/1 - 460037581.

Declaration on Generative AI

The authors are not native English speakers; therefore, we used ChatGPT and Grammarly to assist with grammar corrections, spell-checking, and rewriting certain passages for clarity and conciseness.

References

[1] A. Crooks, A. Croitoru, A. Stefanidis, J. Radzikowski, # earthquake: Twitter as a distributed sensor system, Transactions in GIS 17 (2013) 124–147.
[2] A. Padmanabhan, S. Wang, G. Cao, M. Hwang, Z. Zhang, Y. Gao, K. Soltani, Y. Liu, Flumapper: A cybergis application for interactive analysis of massive location-based social media, Concurrency and Computation: Practice and Experience 26 (2014) 2253–2265.
[3] W. Hobbs, N. Lajevardi, Effects of divisive political campaigns on the day-to-day segregation of arab and muslim americans, American Political Science Review 113 (2019) 270–276.
[4] R. Jurdak, K. Zhao, J. Liu, M. AbouJaoude, M. Cameron, D. Newth, Understanding human mobility from twitter, PloS one 10 (2015) e0131469.
[5] J. J. Padilla, H. Kavak, C. J. Lynch, R. J. Gore, S. Y. Diallo, Temporal and spatiotemporal investigation of tourist attraction visit sentiment on twitter, PloS one 13 (2018) e0198857.
[6] Y. Hu, R.-Q. Wang, Understanding the removal of precise geotagging in tweets, Nature Human Behaviour 4 (2020) 1219–1221.

[7] S. E. Middleton, G. Kordopatis-Zilos, S. Papadopoulos, Y. Kompatsiaris, Location extraction from social media: Geoparsing, location disambiguation, and geotagging, ACM Transactions on Information Systems (TOIS) 36 (2018) 1–27.
[8] E. Aldana-Bobadilla, A. Molina-Villegas, I. Lopez-Arevalo, S. Reyes-Palacios, V. Muñiz-Sanchez, J. Arreola-Trapala, Adaptive geoparsing method for toponym recognition and resolution in unstructured text, Remote Sensing 12 (2020) 3041.
[9] X. Hu, Z. Zhou, H. Li, Y. Hu, F. Gu, J. Kersten, H. Fan, F. Klan, Location reference recognition from texts: A survey and comparison, ACM Computing Surveys 56 (2023) 1–37.
[10] D. Racek, B. I. Davidson, P. W. Thurner, X. X. Zhu, G. Kauermann, The russian war in ukraine increased ukrainian language use on social media, Communications Psychology 2 (2024) 1.
[11] R. Suwaileh, T. Elsayed, M. Imran, Idrisi-re: A generalizable dataset with benchmarks for location mention recognition on disaster tweets, Information Processing and Management 60 (2023) 103340. URL: https://www.sciencedirect.com/science/article/pii/S0306457323000778. doi:https://doi.org/10.1016/j.ipm.2023.103340.
[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.
[13] E. Shook, V. K. Turner, The socio-environmental data explorer (sede): a social media–enhanced decision support system to explore risk perception to hazard events, Cartography and Geographic Information Science 43 (2016) 427–441.
[14] A. Kruspe, J. Kersten, F. Klan, Detection of actionable tweets in crisis events, Natural Hazards and Earth System Sciences 21 (2021) 1825–1845.
[15] P. Scott, M. K.-F. Bader, T. Burgess, G. Hardy, N. Williams, Global biogeography and invasion risk of the plant pathogen genus phytophthora, Environmental Science & Policy 101 (2019) 175–182.
[16] J. He, W. Shen, P. Divakaruni, L. Wynter, R. Lawrence, Improving traffic prediction with tweet semantics, in: Twenty-third international joint conference on artificial intelligence, Citeseer, 2013.
[17] T. Dasgupta, A. Naskar, R. Saha, L. Dey, Crimeprofiler: Crime information extraction and visualization from news media, in: Proceedings of the international conference on web intelligence, 2017, pp. 541–549.
[18] N. Freire, J. Borbinha, P. Calado, B. Martins, A metadata geoparsing system for place name recognition and resolution in metadata records, in: Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries, 2011, pp. 339–348.
[19] A. V. Aho, M. J. Corasick, Efficient string matching: an aid to bibliographic search, Commun. ACM 18 (1975) 333–340. URL: https://doi.org/10.1145/360825.360855. doi:10.1145/360825.360855.
[20] S. Malmasi, M. Dras, Location mention detection in tweets and microblogs, in: Computational Linguistics: 14th International Conference of the Pacific Association for Computational Linguistics, PACLING 2015, Bali, Indonesia, May 19-21, 2015, Revised Selected Papers 14, Springer, 2016, pp. 123–134.
[21] B. Hughes, T. Baldwin, S. Bird, J. Nicholson, A. MacKinlay, Reconsidering language identification for written language resources, in: N. Calzolari, K. Choukri, A. Gangemi, B. Maegaard, J. Mariani, J. Odijk, D. Tapias (Eds.), Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06), European Language Resources Association (ELRA), Genoa, Italy, 2006. URL: http://www.lrec-conf.org/proceedings/lrec2006/pdf/459_pdf.pdf.
[22] T. Baldwin, M. Lui, Language identification: The long and the short of the matter, in: R. Kaplan, J. Burstein, M. Harper, G. Penn (Eds.), Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Los Angeles, California, 2010, pp. 229–237. URL: https://aclanthology.org/N10-1027.
[23] B. King, S. Abney, Labeling the languages of words in mixed-language documents using weakly supervised methods, in: L. Vanderwende, H. Daumé III, K. Kirchhoff (Eds.), Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Atlanta, Georgia, 2013, pp. 1110–1119. URL: https://aclanthology.org/N13-1131.

[24] I. Balazevic, M. Braun, K.-R. Müller, Language detection for short text messages in social media, 2016. URL: https://arxiv.org/abs/1608.08515. arXiv:1608.08515.
[25] R. Suwaileh, T. Elsayed, M. Imran, H. Sajjad, When a disaster happens, we are ready: Location mention recognition from crisis tweets, International Journal of Disaster Risk Reduction 78 (2022) 103107. URL: https://www.sciencedirect.com/science/article/pii/S2212420922003260. doi:https://doi.org/10.1016/j.ijdrr.2022.103107.
[26] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, 2020. URL: https://arxiv.org/abs/1910.03771. arXiv:1910.03771.
[27] D. Antypas, A. Ushio, F. Barbieri, L. Neves, K. Rezaee, L. Espinosa-Anke, J. Pei, J. Camacho-Collados, Supertweeteval: A challenging, unified and heterogeneous benchmark for social media nlp research, in: Findings of the Association for Computational Linguistics: EMNLP 2023, 2023.
[28] E. Bastianelli, A. Vanzo, P. Swietojanski, V. Rieser, SLURP: A spoken language understanding resource package, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 7252–7262. URL: https://aclanthology.org/2020.emnlp-main.588. doi:10.18653/v1/2020.emnlp-main.588.
[29] J. FitzGerald, C. Hench, C. Peris, S. Mackie, K. Rottmann, A. Sanchez, A. Nash, L. Urbach, V. Kakarala, R. Singh, S. Ranganath, L. Crist, M. Britan, W. Leeuwis, G. Tur, P. Natarajan, Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages, 2022. arXiv:2204.08582.
[30] J. Piskorski, M. Marcińczuk, R. Yangarber, Cross-lingual named entity corpus for Slavic languages, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italy, 2024, pp. 4143–4157. URL: https://aclanthology.org/2024.lrec-main.369.
[31] D. Gomez, Geoparser: A python package for geoparsing text, https://pypi.org/project/geoparser/, 2024. Version 0.1.8.
[32] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, arXiv preprint arXiv:1911.02116 (2019).
[33] B. I. Davidson, D. Wischerath, D. Racek, D. A. Parry, E. Godwin, J. Hinds, D. Van Der Linden, J. F. Roscoe, L. Ayravainen, A. G. Cork, Platform-controlled social media apis threaten open science, Nature Human Behaviour 7 (2023) 2054–2057.

Appendix

A. Aho-Corasick FSA

When processing an input string, the Aho-Corasick FSA moves through the trie according to the characters of the string. If a character does not have a corresponding child in the trie, the algorithm follows the fail link to continue the search. This approach ensures that all potential matches are found efficiently, as the automaton can check for multiple patterns in a single pass through the text. By employing this algorithm, our system can rapidly and accurately identify multiple location mentions, enhancing both the speed and accuracy of the recognition process.

B. Labeling the Ukraine Benchmark Data

The labels for the dataset were assigned by a single annotator with the help of Google Translate, with a second person reviewing the labels to ensure accuracy and consistency. One of the labelers was fluent

in Ukrainian and had a good understanding of Russian.

- The Locations were labeled as is, i.e. including misspellings and typos. Furthermore, in cases where multiple locations are mentioned in the same tweet, all of the mentions were labeled, even when they are referencing the same location.
- To determine geographic coordinates, we utilized OSM information. Generally, when available, the coordinates for the OSM text label were taken. Otherwise, we assigned coordinates based on the centroid of the respective region.