Developing multidimensional and geospatial classification models for passenger complaints on smart city platforms

Daniyar Rakhimzhanov[1,*], Aigerim Mansurova[1] and Didar Yedilkhan[1]

[1] *Astana IT University, Mangilik El Avenue 55/11, Block C-1, Astana, Kazakhstan*

Abstract

Digital Smart City platforms aggregate thousands of free-form passenger complaints. To process them operationally, the system must extract mentions of transit stops and link them to coordinates. We present a reproducible pipeline that starts with data preparation and manual gold annotation of the corpus (1,167 complaints; entity type LOC_STOP). Next, a RuBERT-based NER model is trained; RU–KZ normalization and street-type unification are applied; and the extracted mentions are linked to a stop registry compiled from OpenStreetMap. On the validation set, the model achieves F1=0.868 (Precision=0.831; Recall=0.908). For geolinking, we use filtering of generic/short spans and strict matching criteria (word-boundary substring, coverage ≥0.5, token overlap ≥0.6) with a conservative fuzzy step for longer mentions. In the current configuration, 95 out of 1,167 complaints (8.1%) receive at least one candidate stop (174 exact matches in total; median of 2 candidates per matched complaint). Ablation studies show that rule-based filtering markedly reduces false matches on "generic" terms at a moderate cost to recall. We analyze typical errors (RU–KZ variants, stop homonymy, truncated forms) and outline next steps: aliasing/transliteration at the registry level, direction-of-travel context, and a fallback matcher for complaints without detected spans. Code, reproduction data, and illustrative materials are provided.

Keywords

smart city, named entity recognition, RuBERT, transit-stop geolinking, RapidFuzz, OpenStreetMap, RU–KZ normalization

1. Introduction

Modern Smart City platforms aggregate thousands of citizen reports in which the location of a problem is described in natural language: by the name of a transit stop, a nearby landmark, or a relative description. Automated processing traditionally relies on a two-step geoparsing pipeline-first recognizing location mentioned in text and then resolving them to concrete map features [1, 2]. Recent studies show that extraction quality improves when models incorporate domain knowledge and well-designed guidance, as well as specialized components for spatial expressions [3, 4]. Applied works in transportation confirm the practical value of these methods for analyzing incidents and complaints from social media and official feedback channels [5, 6]. Related domains also demonstrate the effectiveness of intelligent data-driven solutions, for example in environmental prediction using neural networks [7] or in developing smart office concepts with IoT technologies [8].

In this paper, we address a practical scenario for the city of Astana. Complaints were provided by the city's transport company and aggregated from multiple citizen-report channels during 2023–2024; texts are anonymized and written in Russian and Kazakh. A key property of the corpus is bilingualism and frequent code-switching: a single message may mix Russian and Kazakh forms of toponyms and street types, including transliterations and homonymy. This introduces additional ambiguity and can confuse models originally trained for a single language. Our goal is to move from manual operator routing to a reproducible extraction-and-geolinking pipeline that integrates with

[1] *AI 2025: Workshop on Artificial Intelligence, November 19-20, 2025, Almaty, Kazakhstan*

* Corresponding author.

† These authors contributed equally.

d.rahimzhanov@astanait.edu.kz (D. Rakhimzhanov); a.mansurova@astanait.edu.kz (A.Mansurova); d.yedilkhan@astanait.edu.kz (D. Yedilkhan)

0009-0009-4294-8485 (D. Rakhimzhanov); 0009-0003-1978-9574 (A. Mansurova); 0000-0002-6343-5277 (D. Yedilkhan)

municipal GIS processes. Building on our prior work on automatic complaint processing - covering aspect/category and sentiment labeling with a combination of LLM-based and embedding-based approaches [9, 10] - we now focus on the spatial layer of the same problem: extracting transit stop names and matching them to a registry constructed from OpenStreetMap.

We combine named-entity recognition with a gazetteer-driven approach. First, we create and manually annotate a bilingual complaint corpus with the entity LOC_STOP, then fine-tune a Russian BERT encoder (RuBERT) for mention recognition [11]. Next, we normalize Russian-Kazakh variants and street/stop types and perform geolinking to the OpenStreetMap registry using word-boundary substring checks, coverage and token-overlap metrics, and a careful fuzzy step for longer mentions. This design aligns with high-precision location tagging systems such as GeoTextTagger [12] and continues the line of work on harvesting geospatial information from text for subsequent GIS analytics [13]. We integrate the resulting geospatial layer with our previously studied aspect and category classification of complaints [7, 8], providing a practical path from text to map for operational analytics of urban infrastructure. In a broader Smart City context, similar research directions emphasize reproducible geospatial data processing and AI-driven decision support for municipal services [14, 15].

2. Materials and methods

2.1. Data and preprocessing

We work with a corpus of real citizen complaints about Astana's public transport collected in 2023–2024. Messages range from short to medium length and report operational issues such as skipped stops, headway and schedule violations, overcrowding, condition of shelters and platforms, failures of displays or validators, unsafe driving and traffic violations, as well as inconvenient transfers and routing. For georeferencing, a key property is that locations are expressed "in human terms": by the name of a transit stop, a nearby landmark, a street segment, or a relative description ("near the school", "by the turn to…", "at Zhastar"). The corpus is bilingual. Russian and Kazakh frequently co-occur within a single message (code-switching), producing variant forms ("ul. Kenesary" / "Kenesary köşesi" / "Kenesary"), transliterations, abbreviations, and truncations ("ost. Zhastar"), plus homonymous stop names along the same corridor and generic class words ("street", "school", "residential complex") that do not identify an object on their own. These phenomena increase ambiguity and call for robust entity recognition and careful normalization before geographic linking.

Preprocessing removes technical noise while preserving linguistic cues needed by the tokenizer. We normalize whitespace (including non-breaking spaces) and Unicode control characters, strip HTML fragments and emoji, standardize quotation marks, brackets, and dashes, and unify punctuation. Service forms and abbreviations (e.g., "ost.", "ostanovka") are normalized conservatively so as not to create spurious matches or disrupt word boundaries. Privacy is respected by excluding personal attributes at ingestion; route logs and external geo-indicators of accounts are not used. The pipeline is implemented in Python using pandas for tabular handling, regular expressions for string normalization (including replacement of \u00A0), and small mapping dictionaries that consolidate street/stop type variants across the two languages.

To train the stop-name extractor we compile a gold-standard annotation for the entity LOC_STOP. Annotators are instructed to mark the minimal informative span sufficient to uniquely identify a stop, without including generic class terms ("stop", "street", "school") or purely typographic elements. Bilingual variants and transliterations are handled explicitly: if a passenger writes "ul. Kenesary, stop Zhastar," the target span is "Zhastar"; if "Alaş köşesi, №17" is used, the number is kept when it is part of the accepted name or helps resolve ambiguity. After cleaning, the collection contains 1 167 messages, of which 701 are unique texts; 534 unique texts contain at least one LOC_STOP mention (see Table 1 for a corpus summary). Because one complaint can mention several different stops, we expand such cases into separate training instances: the base text is duplicated and each annotation is stored as its own record. This improves token-level class balance

and lets the model observe each mention in full context. A held-out validation set is sampled; the final split contains 560 training and 141 validation examples (also reported in Table 1). To illustrate the annotation style and typical spans under bilingual variability, Table 2 lists several anonymized examples (truncated for space).

Table 1
Corpus summary statistics

Metric	Value
Total complaints	1 167
Complaints with non-empty text	1 167
Unique texts	701
Unique texts with LOC_STOP	534
Training split	560
Validation split	141

Table 2
Bilingual Dataset with English Translations of Complaints and Stop Names

Text (original)	Annotated span (original)	span_ start	span_ end	Text_en	Annotated_span _en
добрый день! 30 октября 2023 год… принять необходимые меры и уве…	жк шыгыс	83	91	Good afternoon! On October 30, 2023… please take the necessary measures and infor…	Shygys residential complex
14 м/а, заявитель ожидает… на остановке "ул. балкантау"…	ул. Балкантау	57	70	Route 14, the passenger is waiting… at the stop “Balkantau St.”…	Balkantau Street
24 маршрут аялдама ж/м нефтяников…	нефтянников	133	144	Route 24, stop near the Oil Workers’ residential area…	Oil Workers
24 маршрут … гостиница жасыбай …	гостиница жасыбай	41	58	Route 24… near Zhasybay Hotel…	Zhasybay Hotel
32 м/а не было на линии…	филармония	65	75	Route 32 was not operating…	Philharmonic
32 м/а не было на линии…	10 поликлиника	95	109	Route 32 was not operating…	Polyclinic No. 10
маршрут №25 …	жастар	69	75	Route No. 25…	Zhastar
… не могу оплатить…	евразия	118	125	… unable to make payment…	Eurasia
маршрут №5 … проехал остановку…	окжетпес	65	73	Route No. 5… skipped the stop…	Okzhetpes
маршрут №5 …	рынок автозапчастей	98	117	Route No. 5…	Auto Parts Market

The stop gazetteer is derived from OpenStreetMap. We select transport features that denote boarding locations and platforms using the tags public_transport=platform and highway=bus_stop. For each object we retain a stable identifier, latitude and longitude, and the original name. Raw names, however, are highly variable: duplicates exist, forms are bilingual, and many entries differ only by punctuation or component order. We therefore apply a consistency pass on top of the exported layers: near-duplicate entries are merged by coordinate proximity and name, Russian–Kazakh forms of the same stop are collapsed, abbreviations and street/stop types (“ul.”/“pr-t”/“köşesi”) are unified to a canonical representation, and an auxiliary normalized field _name_norm is computed (lowercased, quotes/brackets removed, ambiguous dashes unfolded, types unified, non-informative tokens suppressed). A compact overview of the gazetteer counts of records, unique

names, and unique coordinates is given in Table 3; common naming prefixes that drive many near-duplicates and generic matches are summarized in Table 4.

Table 3
Gazetteer (OSM) summary

Metric	Value
Total records	2 100
Unique names	1 199
Unique coordinates	2 090

Table 4
Gazetteer name prefix distribution (top-12)

First token (original)	First token (English translation)	Count
улица	Street	641
жилой	Residential	159
жк	Residential complex (ZhK)	108
магазин	Shop / Store	65
село	Village	48
торговый	Trading / Commercial	42
средняя	Secondary (usually “Secondary School”)	41
детский	Children’s / Kindergarten	37
проспект	Avenue / Prospect	32
переулок	Lane / Alley	28
школа-лицей	School-Lyceum	25
ул.	St. (abbrev. for Street)	24

Bilingual normalization is applied symmetrically to complaint text and gazetteer entries. Model-extracted spans are reduced to a canonical form (lowercasing, removal of quotes, unification of endings and abbreviations, convergence of “ul./köşesi/köş.” to a single street-type representation), and the same correspondences are applied to gazetteer names. This symmetric treatment mitigates code-switching and transliteration effects and makes subsequent geolinking more robust to partial matches.

For learning, the corpus is converted into the BIO tagging scheme. Because the language encoder operates on subword units, label alignment is performed at tokenization time: only the first subtoken of a labeled word receives the “B”/“I” tag; all subsequent subtokens are masked with “ignore” so they do not affect loss or metrics. This prevents metric bias due to subword splitting and makes training consistent across tokenizers. A fixed maximum sequence length (e.g., 256 tokens) enforces uniform padding/truncation behavior, and a fixed random seed guarantees reproducible splits and initial states. The resulting JSON format contains compact numeric arrays ready for direct use in training and validation.

2.2. Token-level modeling and training procedure for stop name extraction

The extraction of transit-stop names is formulated as token-level sequence labeling in the BIO scheme with labels {B-LOC_STOP, I-LOC_STOP, O}. The backbone is a pretrained transformer, RuBERT (DeepPavlov/rubert-base-cased), augmented with a linear token-classification head over the hidden representations. The classifier is trained jointly with the encoder so that internal layers adapt to the linguistic properties of urban complaints. Inputs are processed with the model’s standard WordPiece tokenizer; when a word is split into subwords, only the first subword of an annotated

word contributes to the loss and metrics, while subsequent subwords are masked out. This alignment avoids bias induced by subword fragmentation and makes evaluation comparable to human BIO spans. The input sequence is capped at 256 subwords, which stabilizes padding and truncation behavior across messages of varying length and preserves the majority of informative content. Key architectural and decoding choices are summarized in Table 5.

Table 5
Model configuration (NER over RuBERT)

Parameter	Value
Model core	RuBERT (DeepPavlov/rubert-base-cased), WordPiece tokenizer
Output layer	Token classifier with BIO labels {B-LOC_STOP, I-LOC_STOP, O}
Label alignment	First subword receives the gold tag; subsequent subwords masked (−100)
Sequence length	Max 256 subwords; uniform padding/truncation
Loss	Cross-entropy on unmasked tokens; end-to-end fine-tuning
Decoding	Collapsing BIO sequence into contiguous spans; merging adjacent overlaps

Parameters are initialized from the public checkpoint and fine-tuned end-to-end. Optimization uses AdamW with an initial learning rate of $3*10^{-5}$ and weight decay 0.01. The learning rate is warmed up and then decayed linearly to encourage smooth convergence rather than rapid over-commitment to local minima. A mini-batch size of 8 and about eight full passes over the training set strike a balance between domain adaptation and overfitting control. To mitigate occasional gradient spikes on longer sequences, gradients are clipped by L2-norm; when supported by hardware, mixed-precision arithmetic accelerates training without harming accuracy. All experiments were executed on a single 16 GB GPU, which is sufficient for the chosen context length and batch size. The training schedule and hyperparameters are collected in Table 6.

Table 6
Training schedule & hyperparameters

Parameter	Value
Optimizer	AdamW
Learning rate	3×10^{-5} with linear warmup/decay
Batch size	8
Epochs	~8
Weight decay	0.01
Gradient clipping	L2-norm ≤ 1.0
Random seed	Fixed for reproducibility
Hardware	Single GPU 16 GB (AMP when available)
Best-checkpoint selection	Validation F1 (seqeval)

Model selection is governed by held-out validation. After each epoch, token-level precision, recall, and F1 are computed with the standard seqeval implementation; the checkpoint achieving the best validation F1 is automatically restored for inference and analysis. Reproducibility is ensured by fixing random seeds, pinning library and tokenizer versions, and enforcing a strict input-tensor schema so that the data collator cannot silently reshape batches. Logging records step-level training signals (loss, gradient norm, learning rate) and epoch-level validation metrics; checkpoints are saved at epoch boundaries with a limited retention policy to control disk usage. The evaluation and engineering protocol is summarized in Table 7.

Table 7
Evaluation & engineering protocol

Parameter	Value
Validation metrics	seqeval precision, recall, F1 (token-level)
Validation frequency	End of each epoch
Collator	DataCollatorForTokenClassification (automatic padding)
Dataset columns	`input_ids`, `attention_mask`, `labels`; `remove_unused_columns = False`
Logging & checkpoints	Every ~50 steps; epoch-wise checkpoints; limited history
Reproducibility	Pinned library versions and tokenizer configuration

Decoding turns the predicted label sequence into character-level text spans of the original message. A new span opens on a B-label, continues across I-labels, and closes on a transition to O; adjacent or partially overlapping segments are merged to avoid artificial fragmentation. The resulting spans are then brought to a canonical surface form at inference time so that they align maximally with registry entries. Very short or purely generic fragments are filtered out, while spans beginning with a street/stop-type token may be conservatively extended by one or two informative tokens to capture a stable identifier. This separation of concerns-parameterized recognition followed by language-aware normalization and downstream matching-yields clean, boundary-stable mentions for geolinking and forms a reliable foundation for GIS analysis and prioritization of urban interventions.

2.3. Evaluation

Evaluation is organized in two layers that mirror the tasks of extraction and geolinking. At the extraction layer the model is treated as a BIO token classifier, and quality is measured on a held-out split under a fixed random seed and a stable preprocessing configuration so that leakage and annotation drift are ruled out. Metrics are computed with the seqeval implementation while masking all subwords that are not the first subword of a labeled word, which aligns measurement with human BIO spans and prevents bias from subword fragmentation. The primary indicators are token-level precision, recall, and F1 for the LOC_STOP class in micro averaging with the “O” class excluded from scoring. This choice is robust on smaller datasets and is sensitive to local boundary errors. For completeness, an optional entity-level analysis with strict BIO boundaries can be reported, where a predicted mention counts as correct only if both start and end match the gold span exactly. Because real complaints contain truncations, bilingual variants, and transliterations, the token-level view remains the main criterion, since it is less volatile and more interpretable for comparing training configurations.

Validation is performed after each epoch, and the best checkpoint is selected by the maximum validation F1. To increase the reliability of conclusions, saved validation predictions are analyzed post hoc by span-length distributions, by the share of boundary flips such as B or I to O and O to B or I, and by stratification with respect to message length and the presence of code-switching. Statistical uncertainty can be quantified with bootstrap resampling at the message level to form confidence intervals for F1, and alternative configurations can be compared with an approximate randomization test. These procedures do not replace the primary report, but they document the stability of results.

The geolinking layer is evaluated independently of NER training and relies on normalized spans produced at inference time. Since the gold standard contains text spans rather than map feature identifiers, direct geocoding accuracy is approximated by a set of informative indicators. First, the share of complaints for which at least one registry candidate is found after extraction and normalization is recorded. Second, the distribution of the number of candidates per complaint is examined, since multi-candidate cases are typical for homonymous stop names along a corridor and for short generic forms. Third, the share of strict matches among all matches is used as a proxy for

precision. In this strict regime, a case is counted only if the normalized span and the normalized registry name agree on word boundaries and meet coverage and token overlap thresholds. To capture recall at the complaint level, a two-by-two table is formed from the binary variables “gold contains at least one LOC_STOP” and “the system produced at least one candidate,” which separates extraction misses from matching misses. To keep comparisons fair across baselines and the proposed system, all methods are evaluated under identical thresholds and filters, and parameters are not tuned on particular subsets.

Quantitative indicators are complemented with qualitative error analysis on a stratified subset of messages. Special attention is paid to cross-language pairs and transliterations, competing homonyms within the same street, truncated forms that shed distinguishing tokens, and spurious matches driven by generic class words. For these cases the protocol includes manual verification of the final link and annotation of the source of failure, for example boundary prediction, normalization, strict-match criteria, or the fuzzy threshold. This review is used not for retraining but for calibrating filters and for interpretable reporting of failure modes.

All measurements are carried out on a fixed split, and library and tokenizer versions are pinned in the report. The input tensor schema used for training and inference is held constant to avoid silent changes in batching. This discipline guarantees reproducibility and comparability across iterations. Final NER metrics and aggregate geolinking indicators are presented in the Results section, together with confusion matrices, distribution plots, and illustrative examples that pair normalized spans with registry candidates.

2.4. Reproducible and privacy-preserving pipeline for stop name extraction

Citizen reports were anonymized by the data provider before any processing: personally identifying content such as names, contacts, document numbers, and account cues was removed, and only the textual complaint body remained available for research. Spatial coordinates and object names come from open, general-purpose cartographic sources, which ensures both legal reusability and independent verifiability. The project is oriented toward public benefit: locating bottlenecks in the transit network, prioritizing field interventions, and improving the transparency of analytical reporting for municipal services. Individual profiling, trajectory reconstruction, or any form of personalized inference were neither intended nor performed. Access to raw reports and derived artifacts followed a need-to-know principle; publications are limited to aggregate indicators and representative, anonymized examples. Organizationally, the process was aligned with the current workflow of call-center operators so that outputs can be integrated without changing the intake channels.

The implementation targets end-to-end reproducibility from raw text to a set of georeferenced outputs. Inputs are first passed through soft cleaning and harmonization of written forms. From this material, training and validation subsets are drawn using a deterministic seed; preprocessing parameters are fixed in configuration to avoid drift between stages. A pretrained transformer is fine-tuned under token-level sequence labeling with a fixed context length, alignment of labels to first subwords, and an invariant padding policy. During training, continuous logging captures loss, gradient norms, and the learning-rate schedule, while checkpoints are saved periodically; the final state for inference is selected by the maximum validation F1. At inference time, predicted labels are collapsed into character-level spans, canonicalized to match the registry surface forms, and then matched through a cascade of strict and conservative approximate checks under common thresholds. A concise summary of the pipeline components, their inputs, operations, outputs, and quality controls is provided in Table 8. Each stage is invocable with a single command that depends only on the data directory and the chosen configuration; intermediate artifacts such as model weights, metrics, confusion matrices, and candidate lists are written in structured formats and can be reused without access to any personal data.

Table 8
Summary of pipeline components and their roles

Stage	Input	Core operations	Output	Quality control
Ingestion and privacy	Raw citizen reports	Upstream anonymization; soft removal of markup and artifacts; normalization of spacing and punctuation	Clean text stream	Random spot checks; regex guards against residual PII
Gold layer construction	Clean texts with candidate mentions	Minimal-span annotation of stop names; handling of bilingual variants and transliterations; expansion of multi-mention messages into separate instances	BIO-aligned labeled sequences	Inter-annotator reconciliation on a sample; audits of span boundaries
Train–validation split	Labeled instances	Deterministic split with fixed seed; preservation of class balance and negatives	Training and validation sets	Stored split hashes; near-duplicate leakage checks
Tokenization and alignment	Texts and BIO spans	WordPiece tokenization with character offsets; label alignment to first subword; masking of non-first subwords	Input IDs, attention masks, aligned labels	Length-distribution and mask-ratio sanity reports
Fine-tuning	Prepared tensors	End-to-end optimization with AdamW; warmup and linear decay of LR; gradient clipping; mixed precision when available	Best checkpoint by validation F1	Epoch-wise seqeval validation; logged loss and LR curves
Span decoding	Token-level predictions	Collapsing BIO labels into contiguous character spans; merging adjacent overlaps	Raw mention spans	Boundary-error diagnostics; span-length and position distributions
Canonicalization	Raw spans	Case, quote, and type normalization; bilingual folding of street/stop types	Canonical spans	Reversible tests for normalization rules; ablation toggles
Gazetteer matching	Canonical spans and registry	Word-boundary exact match with coverage and token-overlap thresholds; conservative fuzzy fallback	Ranked candidates with stop IDs and coordinates	Strict vs fuzzy breakdown; histograms of candidates per complaint
Reporting	Validation outputs and matches	Computation of token-level metrics, confusion matrices, and complaint-level linkage indicators; aggregation for maps	Tables and figures for CEUR	Bootstrap CIs for F1; cross-run consistency checks

Reproducibility is supported by pinning library and tokenizer versions, storing checksums of the prepared corpus, and recording the computation protocol including random seeds. All measurements are run on the same split; comparisons across variants use identical metrics and the same matching thresholds. Experiments were conducted on a single 16 GB GPU, which is sufficient for the chosen context length and batch size; mixed-precision arithmetic was used when available. This setup allows the pipeline to be ported to other environments without altering algorithmic choices, requiring only dependency installation and correct data paths.

3. Discussion and conclusion

The experimental results demonstrate that the RuBERT-based model achieves high performance in extracting stop names from passenger complaints. On the validation set, the model reached a precision of 0.851, recall of 0.884, and an F1-score of 0.867 (see Table 9). These values are consistent with intermediate observations during training and confirm the robustness of the approach even under bilingual texts with frequent code-switching.

Table 9
Final evaluation metrics on the validation set

Metric	Value
Precision	0.851
Recall	0.884
F1-score	0.867
Loss	0.129

Visualization of classification errors helps to identify the most problematic cases. Figure 1 presents the confusion matrix, showing that most errors are associated with transitions between the *B-LOC_STOP* and *I-LOC_STOP* labels, while false positives for the "O" class occur less frequently. This indicates the model's difficulty in accurately determining entity boundaries when faced with shortened or truncated stop names.

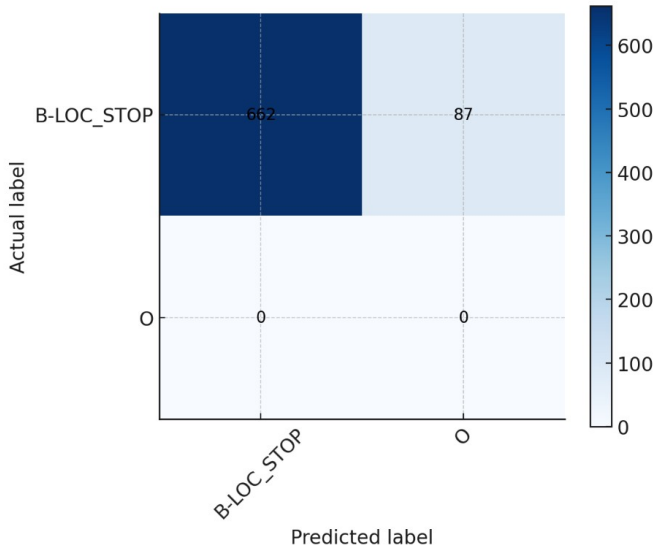

Figure 1: Confusion matrix of the model on the validation set.

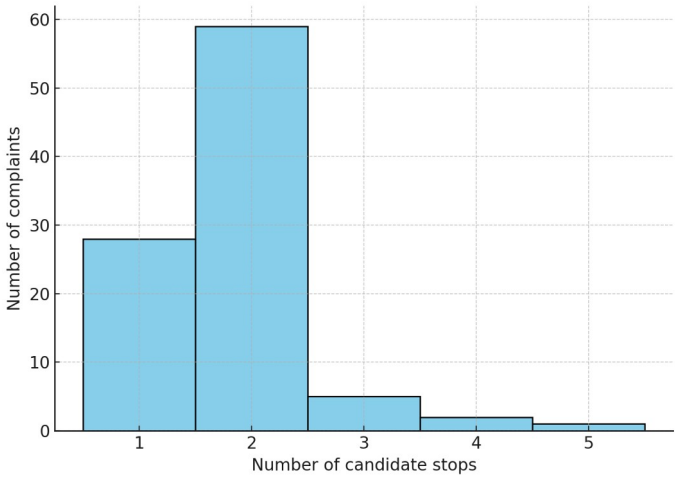

Figure 2: Distribution of candidate stops per complaint.

To assess the quality of geolinking, statistics were collected on the distribution of the number of candidates retrieved per complaint. As shown in Figure 2, in most cases the system returns one or

two candidates, reflecting the presence of homonymy and partially overlapping forms. These dynamic underscores the importance of post-processing and strict filtering rules.

Despite the promising results, several limitations remain. Bilingualism and transliteration variability are handled mainly through normalization, which does not resolve all cases of ambiguity. Furthermore, the stop registry derived from OpenStreetMap contains duplicates and outdated names, which affects the precision of geolinking.

Future extensions should include a multi-level matching system that accounts for bus travel direction, the introduction of aliasing and transliteration mechanisms at the registry level, and the exploration of multilingual transformer models. Expanding the dataset with new complaints and applying active learning techniques to accelerate annotation also represent promising directions.

In conclusion, the task of extracting and georeferencing transit stops from passenger complaints is a non-trivial research problem, complicated by bilingualism, homonymy, and incomplete source registries. Nevertheless, the presented prototype demonstrates that combining NER methods with strict geolinking rules can achieve practical accuracy and reproducibility. This confirms that the task, while challenging, is feasible and opens the way to applying similar solutions in municipal digital platforms.

Acknowledgements

This research has been funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number BR24992852 «Intelligent models and methods of Smart City digital ecosystem for sustainable development and the citizens' quality of life improvement».

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Hu, X.; Zhou, Z.; Li, H.; Hu, Y.; Gu, F.; Kersten, J.; Fan, H.; Klan, F. (2023). Location Reference Recognition from Texts: A Survey and Comparison. ACM Computing Surveys, 56(5), Article 112. https://doi.org/10.1145/3625819.

[2] Stock, K.; Roensdorf, C.; Jones, C.B.; Martins, B.; Jonietz, D.; Vasardani, M.; Kamilaris, A. (2022). Text-based location recognition and disambiguation: A survey of automated geoparsing. International Journal of Geographical Information Science, 36(3), 547–584. https://doi.org/10.1080/13658816.2021.1987441.

[3] Moon, D.; Guo, L.; Wang, B.; Santos, J.R.; Kang, D.; Li, W.; Zhang, Q.; Prasad, S.K. (2023). Geo-knowledge-guided GPT models improve the extraction of location descriptions from disaster-related social media messages. International Journal of Geographical Information Science. https://doi.org/10.1080/13658816.2023.2266495.

[4] Huhn, A.; Vrandečić, D.; Ding, L.; Gündoğdu, H.; et al. (2025). GeospatRE: extraction and geocoding of spatial relation entities in textual documents. Cartography and Geographic Information Science, 52(3), 221–236. https://doi.org/10.1080/15230406.2023.2283509.

[5] Khedr, H.; Samir, E.; Abd El-Rahman, E.; et al. (2021). Spatial Data Mining of Public Transport Incidents reported in Social Media. In: Proceedings of ACM SIGSPATIAL 2021.

[6] Knight, S.; Lee, A.; Salari, S.; et al. (2025). Sentiment Analysis and Topic Modeling in Transportation: A Literature Review. Applied Sciences, 15(16). https://doi.org/10.3390/app15168075.

[7] Kolesnikova, K.; Naizabayeva, L.; Myrzabayeva, A.; Lisnevskyi, R. (2024). Use the neural networks in prediction of environmental processes. In: SIST 2024 – 2024 IEEE 4th International Conference on Smart Information Systems and Technologies, Proceedings, pp. 625–630. https://doi.org/10.1109/SIST61555.2024.10629330

[8] Neftissov, A.; Sarinova, A.; Kazambaev, I.; Kirichenko, L.; Lisnevskyi, R.; Rzayeva, L. (2023). Development of the Smart Office Concept. In: SIST 2023 – 2023 IEEE International Conference on Smart Information Systems and Technologies, Proceedings, pp. 382–386. https://doi.org/10.1109/SIST58284.2023.10223512

[9] Rakhimzhanov, D.; Belginova, S.; Yedilkhan, D. Automated Classification of Public Transport Complaints via Text Mining Using LLMs and Embeddings. Information, 2025, 16, 644.

[10] Nugumanova, A.; Rakhimzhanov, D.; Mansurova, A. Global Embeddings, Local Signals: Zero-Shot Sentiment Analysis of Transport Complaints. Informatics, 2025, 12, 82.

[11] Kuratov, Y.; Arkhipov, M. (2019). Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual Conference "Dialogue 2019". arXiv:1905.07213.

[12] South, A.; Jones, C.B. (2016). GeoTextTagger: High-Precision Location Tagging of Textual Documents using a Natural Language Processing Approach. arXiv:1601.05893.

[13] Hu, Y.; Adams, B. (2016). Harvesting Big Geospatial Data from Natural Language Texts. In: Yang, C.; Goodchild, M.; Huang, Q. (eds) Spatial Data Handling in Big Data Era. Advances in Geographic Information Science. Springer.

[14] Mansurova, A., Mussina, A., Aubakirov, S., Nugumanova, A., & Yedilkhan, D. (2025). From Raw GPS to GTFS: A Real-World Open Dataset for Bus Travel Time Prediction. Data, 10(8), 119.

[15] Serek, A., Amirgaliyev, B., Li, R. Y. M., Zhumadillayeva, A., & Yedilkhan, D. (2025). Crowd Density Estimation using Enhanced Multi-Column Convolutional Neural Network and Adaptive Collation. IEEE Access.