

DangeRoads: Analysis of Road Accident News Articles in the Province of Parma

Michele Tomaiuolo^{1,*}, Gianfranco Lombardo^{1,*}, Elena Azzi¹, Marcello Brianti¹,
Doina Cauliuc¹ and Federico Filice¹

¹Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124 Parma, Italy

Abstract

Road safety monitoring increasingly benefits from the automated analysis of unstructured text sources such as local news articles. In the context of multi-agent smart-city systems, a dedicated RoadSafetyAgent requires a continuously updated, spatially-resolved hazard index to support context-aware routing decisions; this paper addresses the Natural Language Processing pipelines that feed such an agent. Specifically, it presents and jointly evaluates two complementary approaches to extracting road-accident knowledge from Italian-language news articles collected from the online newspaper *ParmaToday* over the period 2016–2022. The first approach employs the *Qwen3-4B-Instruct* large language model (LLM) with a range of prompt engineering strategies to perform full-article binary classification (accident severity, motorcycle/scooter involvement, vulnerable-user involvement) and to extract structured information (date, geolocation), constructing a danger index validated against the Municipality of Parma's OpenData dataset. The second approach benchmarks the same LLM against a fine-tuned Italian BERT XXL model for sentence-level binary labelling of five classes (localisation, time, severity, motorcycle, vulnerable users), and builds a complementary road-hazard grid map with statistical comparison to official data. Results show that dynamic few-shot prompting achieves a macro F1-score of 0.9515 for full-article classification, while sentence-level LLM classification reaches a macro F1 of 0.70 compared to BERT's 0.77, a gap consistent with the inherent mismatch between sentence-level inference and the broader context sensitivity of decoder-only models. Both pipelines demonstrate that unstructured news data can serve as a viable complementary source for territorial road-safety assessment, and constitute the knowledge-extraction backbone of a RoadSafetyAgent operating within a multi-agent smart-city routing architecture.

Keywords

road safety, large language models, prompt engineering, named entity recognition, hazard mapping, smart cities, natural language processing

1. Introduction

Road safety remains a critical public-health concern across Europe and Italy in particular. According to ISTAT data, 173,364 road accidents involving personal injuries occurred in Italy in 2024—a 4.1% increase over the previous year—resulting in 3,030 deaths and 223,853 injuries [1]. The socioeconomic burden is equally staggering: the ACI-ISTAT report estimates the total social cost of road accidents at €22.3 billion for 2023 [2]. Vulnerable road users—pedestrians, cyclists, and motorcyclists—account for 50% of fatalities, while micromobility users are increasingly exposed: in 2024, accidents involving electric bicycles rose by 22% and those involving e-scooters by 66.7% compared to 2023. At the local level, the Province of Parma recorded 1,434 accidents in 2023, producing 1,853 injuries and 24 deaths. Despite being acknowledged as a *virtuous* example of sustainable mobility (with approximately 150 km of cycling infrastructure and active *P-days*), the data reveal that growth in active transport modes is not yet matched by adequate safety outcomes. The December 2024 Italian road-code reform (Bill No. 177/2024) introduced compulsory insurance and helmet requirements for e-scooters and tightened penalties for distracted or impaired driving, reflecting initial institutional attention to these trends.

Smart city initiatives increasingly frame urban mobility as a system-level challenge, requiring the coordinated integration of heterogeneous, continuously updated data streams [19, 20]. Multi-agent systems (MAS) represent a natural and well-established paradigm for this purpose [36, 23, 37, 38, 24]: independent specialised agents—each responsible for a distinct information layer such as traffic flow,

WOA 2026, the 27th Workshop From Objects to Agents, Salerno, June 15-17, 2026

*Corresponding author.



Copyright © 2026 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

weather conditions, public transport disruptions, or road safety—can be orchestrated by a higher-level planning agent to produce integrated, context-aware routing recommendations. Actor-based MAS implementations are particularly suited to this setting, as they support asynchronous, loosely-coupled communication and scale naturally to the distributed, event-driven nature of urban data sources [40, 27]. Agent-based modelling has also proven effective in geographically-grounded simulations of complex urban phenomena, including high-resolution models of population mobility and epidemic spreading [39, 25, 26], demonstrating the viability of spatially-aware agent architectures for territorial monitoring at scale. Within this broader architectural vision, a dedicated RoadSafetyAgent must maintain a spatially-resolved, up-to-date hazard index that peer agents can query to steer users away from dangerous road segments or to trigger proactive re-routing notifications.

Feeding such an agent with timely and accurate data is non-trivial. Official accident registries—such as those published by ISTAT or local police—provide reliable statistics, but they lag in publication, cover only reported incidents, and require dedicated collection infrastructure. Local news articles, by contrast, offer a near-real-time stream of accident reports; however, they are unstructured and heterogeneous. NLP-based approaches have demonstrated strong performance in extracting structured safety information from unstructured accident reports [18, 21, 22], motivating their application to online news as a complementary data source for road-hazard monitoring. More broadly, machine learning and NLP techniques have been successfully applied to knowledge extraction across a wide range of domains [41, 31, 32, 30, 33], consistently demonstrating their ability to handle challenges such as class imbalance, domain-specific vocabulary, and representation learning over heterogeneous data sources that are equally relevant to the road-safety text-mining task addressed here. Recent work further shows that LLMs, including orchestrated multi-step LLM workflows, are emerging as effective tools for structured information extraction and traffic safety analysis in intelligent transportation systems [43, 29, 20, 19].

This paper presents the Natural Language Processing pipelines underlying the RoadSafetyAgent component of a multi-agent smart-city routing system, designed to automatically extract structured road-hazard knowledge from Italian-language news articles. Two complementary pipelines are proposed and evaluated:

1. **Article-level pipeline** applies prompt engineering to the Qwen3-4B-Instruct LLM for *full-article* classification and information extraction, culminating in a geospatial danger index validated against the Municipality of Parma’s 2016–2022 OpenData dataset [3].
2. **Sentence-level pipeline** benchmarks the same LLM against a fine-tuned Italian BERT XXL model for *sentence-level* binary labelling, and constructs a grid-based road-hazard map compared statistically with the same reference data.

Together, they constitute an end-to-end study of LLM applicability to road-accident text mining, spanning prompt design, model comparison, geolocation, and spatial validation, providing the RoadSafetyAgent with a continuously updatable, spatially-resolved hazard index ready for consumption by the orchestrating routing agent.

2. Related Work

2.1. Road-Accident Information Extraction from Text

A work proposed at the 2021 Prognostic and Health Management Society (PHM Society) conference [18] aimed at a similar task, i.e., searching for significant elements to extract from the text of articles regarding road accidents, in natural language. However, the classification was performed manually by experts and not automatically, solely according to the severity of the incident in question, without any relationship between the text and the location of the incidents analyzed, and finally it was based on articles written in English.

A 2024 study, published in *IET Intelligent Transport Systems* [4], addressed binary classification and information extraction from Chinese road-accident news, using deep learning architectures. The work

is interesting, as it relies on task-specific models, rather than general-purpose LLMs. In fact, also our analysis compares task-specific models and BERT in particular, with much larger ad general-purpose models.

A research project, conducted by the Bike Ottawa community in Canada, created a map using OpenStreetMap that describes, through different colors, the level of stress a cyclist experiences while riding a particular road in the Canadian city [14]. Bike Ottawa’s *Collision Map* [7] demonstrates how open data and NLP-derived information can be fused into route-planning tools for cyclists.

Using the open source code [15] provided by Bike Ottawa, Maurizio Napolitano and Matteo Fortini first carried out the same work for their cities, and then extended it throughout Italy to create Traffic-stressed Bikes [16].

Reference Datasets and Spatial Analysis. The Municipality of Parma’s OpenData project [3] provides a publicly available dataset of 5,162 accidents recorded by the local police between 2016 and 2022, including date, time, GPS coordinates, weather conditions, and injury severity—the benchmark used for validation in both pipelines.

At the national level, CYCLANDS [5] aggregates approximately 1.6 million cyclist accidents across Europe, the United States, and Australia.

The Politecnico di Milano’s *Atlas of Cycling Accidents* [6] provides five interactive dashboards based on ISTAT data (2014–2023), illustrating cyclist fatalities and severe injuries across Italy.

3. Background and Technologies

Large Language Models and Transformer Architecture. LLMs represent the current frontier of NLP. Trained on massive corpora via self-supervised objectives, they exhibit strong in-context learning capabilities that can be steered through *prompt engineering* without any task-specific fine-tuning. The dominant architecture is the Transformer [8], comprising self-attention layers that dynamically weight token relationships across the entire input sequence.

BERT [9] is an *encoder-only* Transformer pre-trained with masked language modelling and next-sentence prediction; it excels at classification and extraction tasks when fine-tuned. The GPT family, including Qwen, uses a *decoder-only* architecture with causal (left-to-right) attention, making it suited to generative tasks and capable of following free-form instructions.

Qwen3-4B-Instruct and Qwen3-Embedding-4B. Both pipelines rely on **Qwen3-4B-Instruct** [10], released by Alibaba Cloud in 2025, a dense decoder-only model with 4 billion parameters (3.6 billion excluding embeddings). Its context window of 256,000 tokens allows entire news articles to be processed in a single call. Multilingual support (100+ languages including Italian) makes it directly applicable to the Italian-language corpus. The *Instruct* variant is optimised for direct instruction following with low latency, in contrast to the *Thinking* variant, which externalises chain-of-thought reasoning.

The article-level pipeline additionally employs **Qwen3-Embedding-4B**—a specialised 4B-parameter embedding model that generates contextual vector representations up to 2,560 dimensions with 32,000-token context support. It is used to dynamically select the most semantically similar examples for few-shot prompts.

Italian BERT XXL. At the sentence-level, comparison uses **Italian BERT XXL Cased** [11], a 111-million-parameter encoder pre-trained on an 81 GB Italian corpus (full Italian Wikipedia and the OSCAR dataset). For this study it was fine-tuned for 5 epochs on 132,946 automatically labelled sentences, making it a strong, task-specific baseline.

Prompt Engineering. Prompt engineering encompasses the systematic design of model inputs to elicit desired outputs without weight updates. Three core strategies are employed across both pipelines:

- **Zero-shot:** task described in plain language with no examples. Relies entirely on pre-training knowledge.
- **Static few-shot:** a fixed set of labelled examples is prepended to each prompt. Improves output format consistency and can reduce classification errors on rare classes. Structuring prompts into three components—task definition, few-shot demonstrations, and constrained output format—has been shown effective for NER with LLMs [35], a strategy adopted in the geolocation sub-pipeline of both approaches.
- **Dynamic few-shot:** examples are selected at runtime by computing cosine similarity between the query embedding and a pool of pre-embedded candidates. This preserves the benefits of few-shot learning while adapting the context to each specific input.

Geolocation: NER, Nominatim, and OpenStreetMap. Both pipelines convert textual location mentions into GPS coordinates. The sentence-level pipeline uses **SpaCy it_nerIta_trf** [12], a Transformer-based Italian named entity recognition (NER) model (F1 = 0.915 on its benchmark), to extract location strings from sentences. The article-level pipeline relies on the LLM itself to produce formatted location strings. In both cases, **Nominatim** [13]—OpenStreetMap’s geocoding API—converts location strings to latitude/longitude pairs. Fallback strategies (synonym substitution, combined queries) are implemented to increase geocoding success rates.

4. System Architecture

The analysis and ready availability of up-to-date road safety information are fundamental elements for enabling the widespread adoption of sustainable transportation and commuting habits. The architecture of this use-case scenario is shown in Figure 1. The system follows a hub-and-spoke pattern with a clear separation between the user-facing layer and the monitoring layer.

RoutePlannerAgent acts as the orchestrator: it receives user requests (bike and public transport routing) and aggregates information from five specialized monitoring agents.

TrafficMonitoringAgent, PollutionMonitoringAgent, and WeatherMonitoringAgent provide real-time environmental context that directly affects route quality. PublicTransportMonitoringAgent feeds live schedule and disruption data specifically relevant to multimodal routing. RoadSafetyAgent and RoadAccessibilityAgent are more static in nature, supplying accident records and terrain/slope data respectively — critical for bike routing in particular.

The monitoring agents are independent and do not communicate with each other. All coordination logic lives in RoutePlannerAgent, which synthesizes inputs into a ranked set of route options returned to the user.

Two tiers of monitoring agents are distinguished by color. The teal agents (Traffic, Pollution, Weather) supply real-time, continuously-changing environmental data — they’re queried on every routing request. The blue agents (PublicTransport, RoadSafety, RoadAccessibility) tend to update on a slower cadence: schedules, accident records, and slope data change less frequently and could be partially cached.

RoutePlannerAgent is the only agent with user-facing responsibility. All other agents are internal services. This is a good design boundary — if you later want to add a CyclingInfrastructureAgent or a ParkingAgent, you plug it into the hub without touching anything else.

Bidirectional arrows reflect that the planner both queries the monitoring agents and can receive push notifications (e.g., a sudden road closure from RoadSafetyAgent triggering a proactive re-route suggestion).

4.1. Road safety analysis

Both pipelines for road safety analysis follow a Knowledge Discovery in Databases (KDD) workflow: data collection, preprocessing, transformation, classification/extraction, and analysis/visualisation. Figure 2 summarises the shared and divergent components.

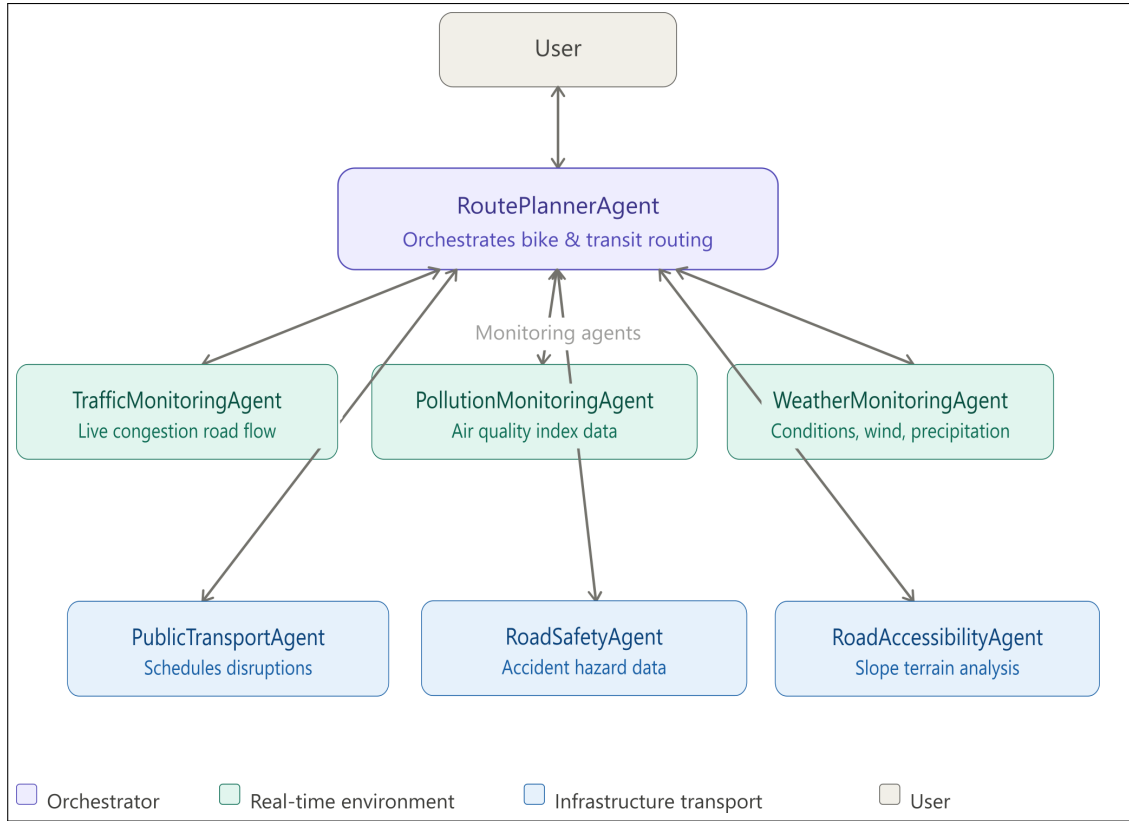


Figure 1: Architecture of the Multi-Agent System.

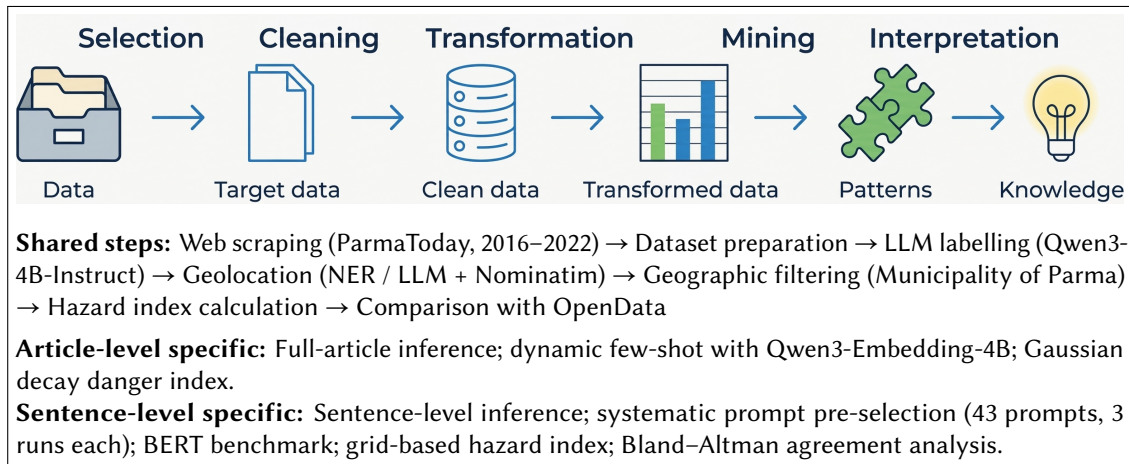


Figure 2: High-level architecture shared by both pipelines.

4.2. Dataset

This research work, in particular, is focused on the analysis of road safety data, to feed the **RoadSafetyAgent** in the more general use case scenario. Both pipelines scrape accident articles from the *ParmaToday* online newspaper using requests and BeautifulSoup4. The article-level pipeline extracts 1,600 full articles (2016–2022), setting aside 200 for example pools and 400 for testing. The sentence-level pipeline collects 1,777 articles, splits them into 10,868 sentences, and uses 900 manually labelled sentences for prompt selection and BERT comparison.

4.3. Classification Labels

The two pipelines share an overlapping but not identical label set (Table 1).

Table 1

Classification labels used in each pipeline.

Label	Article	Sentence
Severity (<i>grave</i>)	☑	☑
Motorcycle/scooter (<i>moto</i>)	☑	☑
Vulnerable users (<i>utenti deboli</i>)	☑	☑
Localisation	—	☑
Time	—	☑
Multiple accidents (<i>multipli</i>)	☑	—
Intermediate point (<i>punto_intermedio</i>)	☑	—

4.4. Article-level analysis: Prompt Configurations

The article-level analysis systematically investigates six prompt configurations organised along two dimensions: *combined vs. separated tasks* and *zero-shot vs. few-shot*.

Combined zero-shot A single prompt requests all seven output fields.

Combined static few-shot Eight manually selected examples appended to the combined prompt.

Combined dynamic few-shot The five most similar examples from a 200-article pool are selected by cosine similarity of Qwen3-Embedding-4B vectors.

Separated prompts Independent prompts for location, date, and labels. For labels, three sub-variants are tested: zero-shot, static few-shot (8 examples, balanced across label combinations), and dynamic few-shot (balanced joint selection across all eight $\{0, 1\}^3$ combinations).

Per-label dynamic few-shot Each label classified independently with 4 positive and 4 negative dynamically selected examples.

Output format is always JSON. Temperature is set to 0 for classification (deterministic sampling) and slightly higher for location extraction. All experiments run on Google Colab with an NVIDIA L4 GPU (24 GB).

4.5. Sentence-level analysis: Prompt Pre-selection and Final Selection

For the sentence-level analysis, a two-stage prompt selection process is conducted. In the pre-selection stage, 43 prompts are evaluated on 200 sentences using Qwen2.5-1.5B-Instruct on Google Colab, each run three times to estimate standard deviation. Prompts span four combinations of *multilabel/monolabel* and *zero-shot/few-shot*, with 2, 4, 6, and 8 examples tested for few-shot variants. The eight best-performing prompts proceed to final selection, re-run on the full 900-sentence dataset using Qwen3-4B-Instruct on the University of Parma’s HPC cluster (NVIDIA Tesla P100, 24 GB, managed via SLURM).

4.6. Hazard Index Formulations

Article-level (Gaussian decay):

$$p_x = \frac{1}{n} \sum_{i=1}^n p_i \cdot e^{-d_{ix}^2} \quad (1)$$

where $p_i \in \{1.0, 0.3\}$ is the severity weight of accident i and d_{ix} is its Euclidean distance from evaluation point x . Accidents labelled *grave* receive weight 1.0; others receive 0.3.

Sentence-level (grid cell sum): For each $0.5 \text{ km} \times 0.5 \text{ km}$ grid cell, the index is

$$H_c = \sum_{i \in c} w_i, \quad w_i = \begin{cases} 1.0 & \text{if severe} \\ 0.5 & \text{otherwise} \end{cases} \quad (2)$$

After computation, Tukey IQR-based outlier capping ($k = 3$) and Min-Max normalisation to $[0, 1]$ are applied before comparison.

5. Implementation Details

Data Preprocessing In both pipelines, scraped articles are parsed via regular expressions to extract publication dates and body text. The article-level analysis stores 1,600 articles in a single CSV file. The sentence-level analysis splits articles into sentences and produces a six-column CSV (article ID, sentence, five binary labels initialised to zero) compatible with the BERT labelling format used in prior work.

For the embedding-based components, the article-level analysis pre-computes Qwen3-Embedding-4B embeddings for the 200-article example pool:

1. *Tokenisation* with truncation at 512 tokens.
2. *Mean pooling* over non-padding token embeddings, using the attention mask to exclude padding positions.
3. *L2 normalisation*: $\hat{v} = v / \|v\|_2$, enabling cosine similarity to be computed as a plain dot product.

Embeddings are persisted as PyTorch .pt tensors; example metadata is stored in pickle .pkl files.

Geolocation pipeline for article-level analysis. The LLM is instructed to return location strings in the format `StreetName City` (no commas), with special rules for motorways (A1 Parma) and intersections (two endpoints separated by a comma). A dedicated *intermediate_position* binary flag triggers midpoint interpolation. Progressive fallback queries—replacing *piazzale* with *piazza*, removing qualifiers—recover many initially unmatched locations.

Geolocation pipeline for sentence-level analysis. SpaCy NER extracts location entities, enriched by RegEx-based prefix concatenation (*via, strada, piazza* + entity $\rightarrow$ full toponym). Post-processing filters remove hospital names and redundant *Parma* occurrences. A cascading Nominatim strategy (individual entities $\rightarrow$ combined query) maximises geocoding recall. The Shapely library defines a 25-vertex polygon of the municipal boundary; only accidents falling inside are retained for final comparison (1,777 $\rightarrow$ 612 accidents).

Spatial Comparison Both pipelines compare news-derived and official hazard indices. The article-level analysis uses the Pearson correlation coefficient on a rasterised grid; the sentence-level analysis applies both Pearson correlation and Bland–Altman analysis on 438 matched grid points where at least one source is non-zero. Heatmaps are generated with *folium* and exported as interactive HTML.

6. Results

Article-level analysis: Classification and Extraction Performance. Table 2 reports macro and weighted F1 scores for static prompt configurations on the 400-article test pool.

Dynamic few-shot results are reported in Table 3. The best dynamic configuration (balanced label classification) achieves a macro F1 of 0.9515, only 0.07 points below the best static configuration.

Key observations: (i) task separation consistently benefits classification; (ii) static few-shot combined prompts cause near-complete date-extraction failure (1% accuracy) because the model copies the year from examples; (iii) dynamic few-shot date extraction (3 similar examples) reaches 83.45% correct dates; (iv) zero-shot location extraction yields 53.48% exact matches and 76.19% within 5 km.

Table 2

Article-level pipeline – static prompt configurations (test pool, $n = 400$). ZS = zero-shot; FS = static few-shot; labels = task-separated prompt.

Configuration	F1-Macro	F1-Wtd	F1 _{grave}	F1 _{moto}	F1 _{deboli}
ZS combined	0.9385	0.9551	0.9493	0.9560	0.9103
FS combined	0.9307	0.9633	0.9778	0.9157	0.8987
ZS labels	0.9382	0.9471	0.9241	0.9873	0.9032
FS labels	0.9588	0.9742	0.9766	0.9872	0.9125

Table 3

Article-level pipeline – dynamic few-shot configurations.

Configuration	F1-Macro	F1-Wtd	F1 _{grave}	F1 _{moto}	F1 _{deboli}
Combined	0.8899	0.9115	0.8752	0.9202	0.8743
Balanced joint	0.9515	0.9725	0.9810	0.9872	0.8862
Per-label	0.9471	0.9691	0.9762	0.9872	0.8780

Sentence-level analysis: Prompt Selection. Table 4 summarises final-selection results for eight prompts on the 900-sentence dataset with Qwen3-4B-Instruct.

Table 4

Sentence-level pipeline – final prompt selection (Qwen3-4B-Instruct, $n = 900$). ML = multilabel; Mono = monolabel; ZS = zero-shot; FS = few-shot; ex = examples per label.

Prompt (type)	$\overline{\text{F1-Macro}}$	$\overline{\text{F1-Wtd}}$	$\text{CV}\%_{\text{Macro}}$	$\text{CV}\%_{\text{Wtd}}$
1 (ML-ZS)	0.67 ± 0.02	0.73 ± 0.02	2.3	2.8
10 (Mono-ZS)	0.65 ± 0.02	0.78 ± 0.02	3.8	2.5
6 (Mono-FS-2ex)	0.68 ± 0.03	0.80 ± 0.02	4.2	2.8
7 (Mono-FS-4ex)	0.69 ± 0.03	0.80 ± 0.02	3.7	2.6
8 (Mono-FS-6ex)	0.65 ± 0.03	0.78 ± 0.04	4.8	4.7
9 (Mono-FS-8ex)	0.61 ± 0.03	0.73 ± 0.04	5.0	5.0
11 (ML-FS-8ex)	0.67 ± 0.03	0.74 ± 0.04	3.9	5.0
2 (ML-FS-4ex)	0.67 ± 0.03	0.74 ± 0.03	3.8	4.2

Prompt 7 (monolabel few-shot, 4 examples per label) is selected for downstream tasks. A notable finding is that performance *decreases* monotonically as the number of few-shot examples increases beyond 4, suggesting a noise or distraction effect for this sentence-level task with a 4B model. All CV% values remain below 5%, confirming high output stability.

Sentence-level analysis: LLM vs. BERT Comparison. Table 5 compares Qwen3-4B-Instruct (best prompt) against fine-tuned Italian BERT XXL on the 900-sentence dataset.

BERT maintains a consistent advantage: on average, 7 percentage points in macro F1 and 5 points in weighted F1. The gap is smallest for *Severity* (only 2 points macro), and larger for time-sensitive and localisation labels where bidirectional context is most beneficial. Notably, Qwen matches or slightly exceeds BERT on *Motorcycle* macro F1 (0.61 vs. 0.67)—a highly imbalanced class with only 54 positive instances out of 900—suggesting that the LLM’s broader world knowledge partially compensates for fine-tuning on imbalanced data for rare categories.

Spatial Validation: Hazard Maps. Table 6 summarises the spatial comparison metrics for both pipelines.

Both correlations are statistically significant ($p < 0.05$). The higher correlation (0.79) at the article-level is attributable to several complementary factors: (i) full-article context produces more accurate

Table 5

Per-label and average macro/weighted F1: Qwen3-4B-Instruct vs. Italian BERT XXL.

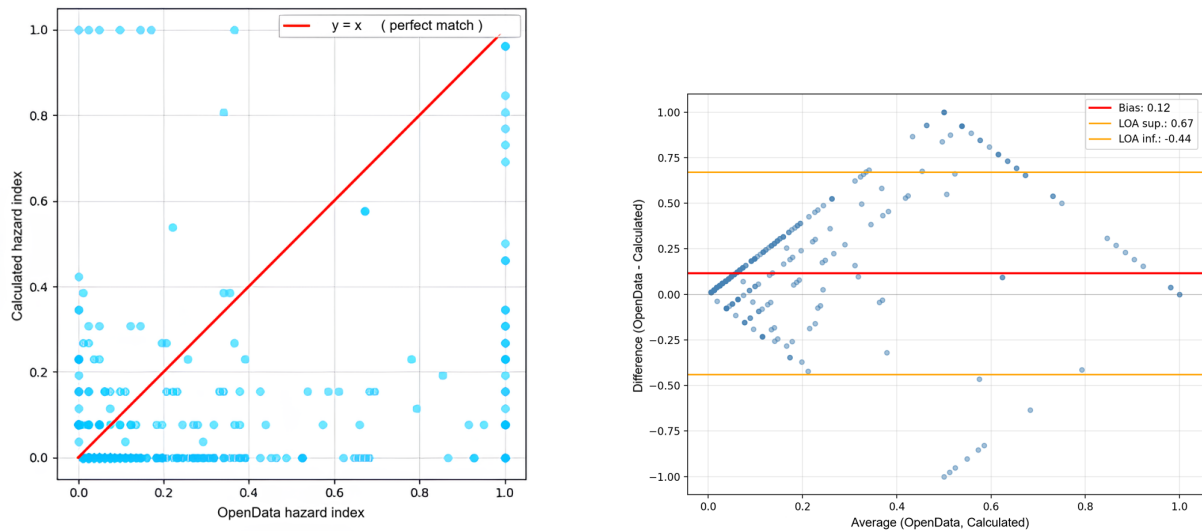
Label	Qwen3-4B-Instruct		BERT XXL	
	Macro F1	Wtd F1	Macro F1	Wtd F1
Localisation	0.72 ± 0.04	0.73 ± 0.03	0.82	0.82
Time	0.74 ± 0.01	0.80 ± 0.01	0.89	0.91
Severity	0.72 ± 0.01	0.75 ± 0.01	0.74	0.77
Motorcycle	0.61 ± 0.06	0.89 ± 0.03	0.67	0.88
Vulnerable users	0.65 ± 0.02	0.84 ± 0.02	0.75	0.86
Average	0.70 ± 0.01	0.80 ± 0.01	0.77	0.85

Table 6

Spatial correlation between news-derived and OpenData hazard maps. B-A = Bland–Altman analysis; LOA = limits of agreement.

Pipeline	Pearson r	Bias (B-A)	Points outside 95% LOA
Article-level	0.79	—	—
Sentence-level	0.40	0.12	8.4%

label predictions; (ii) the LLM directly extracts location strings, avoiding the NER error chain; (iii) the Gaussian decay index distributes accident influence across the neighbourhood, reducing sensitivity to exact coordinate precision. The lower correlation (0.40) at the sentence-level reflects the compounding effects of sentence-level labelling errors and NER-based geolocation, as well as the 8:1 imbalance in accident counts between the 5,101-record OpenData and the 612 geolocated news-article records. The Bland–Altman analysis indicates a systematic underestimation bias of 0.12 from news sources, consistent with the media selection effect: newspapers predominantly report the most notable accidents, underrepresenting minor events.

**Figure 3:** Sentence-level pipeline graphs, Scatter plot (left), Bland-Altman plot (right).

A qualitative evaluation of the two Hazard Maps, derived from the Sentence-level pipeline process, has been created to visually show the differences in hazard index detection between the data resulting from ParmaToday newspaper articles and the OpenData dataset.

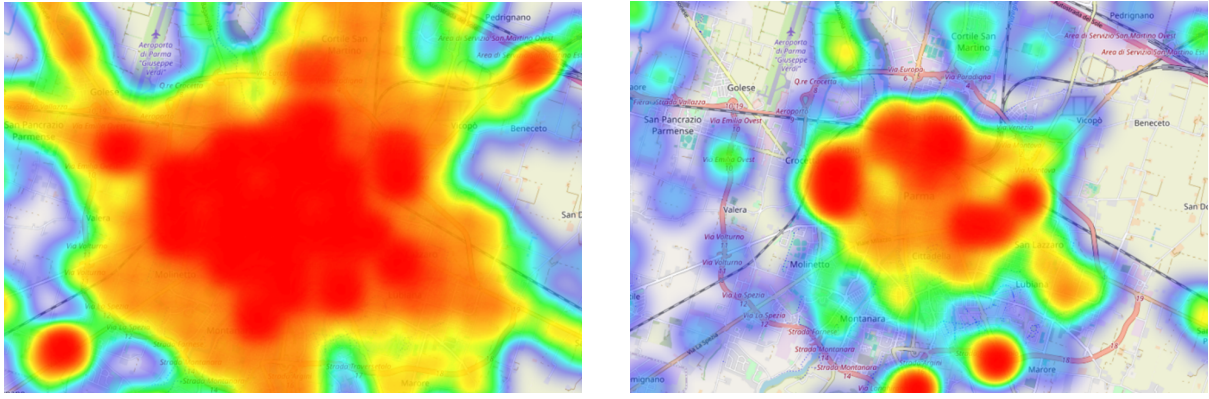


Figure 4: Hazard Maps based on OpenData dataset from ParmaToday (left), from sentence-level pipeline (right).

7. Discussion

Prompt Engineering Findings. Across both pipelines, several robust prompt engineering conclusions emerge:

1. **Task decomposition is beneficial.** Separating classification from information extraction, and decomposing multi-class prompts into per-label or per-task components, consistently improves both accuracy and output reliability. Combined prompts introduce interference between tasks, most dramatically causing the model to copy example dates when date extraction is bundled with classification.
2. **Few-shot has diminishing returns.** For the 4B model, 4 examples per label represents the sweet spot in monolabel regime of the sentence-level analysis. Beyond this, prompt length increases without commensurate gain, and may even degrade performance by saturating the context with less relevant material. The optimal number appears correlated with model capacity.
3. **Dynamic few-shot preserves most of the benefit.** The article-level analysis demonstrates that replacing fixed examples with semantically similar dynamically selected ones achieves macro F1 within 0.07 points of the best static configuration, while offering automation advantages at scale.
4. **Temperature matters.** The sentence-level analysis shows that slightly lower temperature ($T = 0.6$, $\text{top_p} = 0.2$) relative to model defaults provides modest gains for classification, consistent with the desire for deterministic, structured JSON output.

LLM vs. Fine-tuned Model. The BERT advantage (7 pp macro F1) is contextualised by three key asymmetries: (i) BERT was fine-tuned on 132,946 sentences, whereas Qwen sees only the prompt; (ii) BERT is an encoder-only model, architecturally suited to classification; (iii) BERT operates on individual sentences with bidirectional attention, while Qwen’s decoder processes them left-to-right, limiting future context. Full-article inference outperforms sentence-level inference for LLMs, which strongly suggests that the observed gap would narrow substantially under a more natural task formulation. Using a 32B or 72B Qwen model—which has been explicitly excluded to maintain fair computational parity with BERT—would likely close or reverse the gap. This result aligns with the broader NLP literature: fine-tuned encoder-only models consistently outperform zero-shot or few-shot prompted generative models on structured classification tasks, even when the latter are substantially larger [34].

Media Bias in Hazard Estimation. A fundamental limitation shared by both pipelines is the *media selection effect*: online newspapers systematically over-report severe or unusual accidents and under-report routine minor collisions. This produces a hazard map that qualitatively agrees with official data on *where* accidents concentrate, but understates *how many* minor incidents occur. The Bland–Altman

bias of 0.12 quantifies this effect: on average, the news pipeline assigns 12 normalised index units less than official data. Future work should consider weighting correction factors derived from the known ratio of media-reported to total accidents.

8. Conclusions and Future Work

This paper has presented two complementary LLM-based pipelines for mining road accident knowledge from local Italian-language news articles. The article-level pipeline achieves macro F1 of 0.9515 for three-class binary classification and 83% accuracy for both location and date extraction, with a spatial correlation of 0.79 with official data. The sentence-level pipeline reaches macro F1 of 0.70 with Qwen3-4B-Instruct (vs. 0.77 for fine-tuned BERT XXL), and a spatial correlation of 0.40, with a systematic underestimation bias of 0.12.

Together, these results demonstrate that:

- General-purpose LLMs can perform road-accident classification and information extraction at competitive accuracy without task-specific training, particularly at the full-article level.
- Dynamic few-shot prompting with semantic similarity-based example selection achieves near-static few-shot performance with greater flexibility.
- News-derived hazard maps qualitatively reproduce the spatial structure of official data, but quantitative agreement is limited by media selection effects and NLP pipeline errors.
- Fine-tuned encoder models (BERT) retain a meaningful advantage for sentence-level binary classification, but this gap is likely reducible by using larger LLMs or full-article inference.

Future directions include: extending scraping to additional sources (institutional press releases, social media) to reduce media bias; integrating road-network graph representations to replace imprecise point-based geolocation; experimenting with larger Qwen models (32B+) to assess the fine-tuning parity threshold; adding temporal and causal label extraction (accident cause, weather conditions) to enable richer trend analyses; and evaluating transferability of the pipeline to other Italian provinces lacking official accident registries.

9. Acknowledgments

The Municipality of Parma is acknowledged for making the OpenData road accident dataset publicly available.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude Sonnet 5 in order to: Grammar and spelling check; Text Translation. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] ISTAT, "Road Accidents in Italy – 2024," <https://www.istat.it/comunicato-stampa/incidenti-stradali-in-italia-2024/>, 2025.
- [2] ACI-ISTAT, "Report on Road Accidents in 2023," <https://aci.gov.it/comunicati-stampa/aci-istat-report-degli-incidenti-stradali-2023/>, 2024.
- [3] Municipality of Parma, "Overall Road Accidents from 2016 to 2022," <https://opendata.comune.parma.it/dataset/incidenti-stradali-2022>, 2022.

- [4] Y. Ling, Z. Ma, X. Dong, and X. Weng, "A Deep Learning Approach for Robust Traffic Accident Information Extraction from Online Chinese News," *IET Intelligent Transport Systems*, vol. 18, no. 10, pp. 1847–1862, 2024. <https://doi.org/10.1049/itr2.12493>
- [5] CYCLANDS Project, "A Global Dataset of Cyclist Accidents," <https://cyclands.sourceforge.io/>, 2022.
- [6] Politecnico di Milano – CRAFT, "Atlas of Cycling Accidents in Italy," <https://craft.dastu.polimi.it/it/articles/15>, presented 21 October 2024.
- [7] Bike Ottawa, "Interactive Maps: Collision Map & Cycling Level of Traffic Stress," <https://maps.bikeottawa.ca/>.
- [8] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2018.
- [10] B. Yang et al., "Qwen3 Technical Report," arXiv:2505.09388, 2025.
- [11] S. Schweter, "Italian BERT and ELECTRA Models," Zenodo, November 2020. <https://zenodo.org/records/4263142>.
- [12] BullMount, "it_nerIta_trf – HuggingFace," https://huggingface.co/bullmount/it_nerIta_trf.
- [13] OpenStreetMap / Nominatim, "Nominatim Geocoding API," <https://nominatim.org/>.
- [14] BikeOttawa. Level of stress map for ottawa, 2023. <https://maps.bikeottawa.ca/lts/>.
- [15] BikeOttawa. maps.bikeottawa.ca-frontend, 2023. <https://github.com/BikeOttawa/maps.bikeottawa.ca-frontend>.
- [16] Maurizio Napolitano. Lo stress dei percorsi ciclabili 2021. <https://medium.com/tantotanto/lo-stress-dei-percorsi-ciclabili-ee7573ec8284>.
- [17] Ipsos. Cycling across the world, a 30-country global advisor survey. Technical report, Ipsos, may 2022. <https://11nq.com/LrkK7>.
- [18] Dario Valcamonico, Piero Baraldi, Francesco Amigoni, and Enrico Zio. A natural language processing method for the identification of the factors influencing road accident severity. In *Proceedings of the 6th European Conference of the Prognostics and Health Management Society*, volume 6, page 12, Jun 2021. <http://papers.phmsociety.org/index.php/phme/article/view/2899>.
- [19] A. Kalyuzhnaya, S. Mityagin, E. Lutsenko, A. Getmanov, Y. Aksenkin, K. Fatkhiev, K. Fedorin, N. O. Nikitin, N. Chichkova, V. Vorona, and A. Boukhanovsky, "LLM Agents for Smart City Management: Enhancing Decision Support Through Multi-Agent AI Systems," *Smart Cities*, vol. 8, no. 1, p. 19, 2025. <https://doi.org/10.3390/smartcities8010019>
- [20] D. Zhang, H. Zheng, W. Yue, and X. Wang, "Advancing ITS Applications with LLMs: A Survey on Traffic Management, Transportation Safety, and Autonomous Driving," in *Rough Sets (IJCRS 2024)*, ser. Lecture Notes in Computer Science, vol. 14840, Springer, Cham, 2024, pp. 295–309. https://doi.org/10.1007/978-3-031-65668-2_20
- [21] A. Ashutosh and S. Chand, "Dataset on fatal road traffic crash attributes extracted via natural language processing of online media articles in India," *Data in Brief*, vol. 60, p. 111578, 2025. <https://doi.org/10.1016/j.dib.2025.111578>
- [22] Y. Ling, Z. Ma, X. Dong, and X. Weng, "A deep learning approach for robust traffic accident information extraction from online Chinese news," *IET Intelligent Transport Systems*, vol. 18, no. 10, pp. 1847–1862, 2024. <https://doi.org/10.1049/itr2.12493>
- [23] G. Angiani, P. Fornacciari, G. Lombardo, A. Poggi, and M. Tomaiuolo, "Actors Based Agent Modelling and Simulation," in *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, ser. Communications in Computer and Information Science, vol. 887, Springer, Cham, 2018, pp. 443–455. https://doi.org/10.1007/978-3-319-94779-2_38
- [24] G. Lombardo, P. Fornacciari, M. Mordonini, M. Tomaiuolo, and A. Poggi, "A Multi-Agent Architecture for Data Analysis," *Future Internet*, vol. 11, no. 2, p. 49, 2019. <https://doi.org/10.3390/fi11020049>
- [25] M. Pellegrino, G. Lombardo, S. Cagnoni, and A. Poggi, "High-Performance Computing and ABMS for High-Resolution COVID-19 Spreading Simulation," *Future Internet*, vol. 14, no. 3, p. 83, 2022. <https://doi.org/10.3390/fi14030083>
- [26] G. Lombardo, M. Pellegrino, M. Tomaiuolo, S. Cagnoni, M. Mordonini, M. D. L. Giacobini, and

- A. Poggi, "Fine-Grained Agent-Based Modeling to Predict Covid-19 Spreading and Effect of Policies in Large-Scale Scenarios," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2052–2062, 2022. <https://doi.org/10.1109/JBHI.2022.3160243>
- [27] M. Pellegrino, G. Lombardo, and A. Poggi, "Towards Digital Twins in Healthcare: Optimizing Operating Room and Recovery Room Dynamics," *Procedia Computer Science*, vol. 246, pp. 4732–4740, 2024. <https://doi.org/10.1016/j.procs.2024.09.338>
- [28] G. Lombardo and A. Poggi, "ActorNode2Vec: An Actor-based solution for Node Embedding over large networks," *Intelligenza Artificiale*, vol. 14, no. 1, pp. 103–114, 2020. <https://doi.org/10.3233/IA-190038>
- [29] G. Trimigno, G. Lombardo, M. Tomaiuolo, S. Cagnoni, and A. Poggi, "LLMs in Staging: An Orchestrated LLM Workflow for Structured Augmentation with Fact Scoring," *Future Internet*, vol. 17, no. 12, p. 535, 2025. <https://doi.org/10.3390/fi17120535>
- [30] G. Ranjbaran, D. Reforgiato Recupero, G. Lombardo, and S. Consoli, "Leveraging augmentation techniques for tasks with unbalancedness within the financial domain: a two-level ensemble approach," *EPJ Data Science*, vol. 12, no. 1, p. 24, 2023. <https://doi.org/10.1140/epjds/s13688-023-00402-9>
- [31] S. Cagnoni, L. Cozzini, G. Lombardo, M. Mordonini, A. Poggi, and M. Tomaiuolo, "Emotion-based analysis of programming languages on Stack Overflow," *ICT Express*, vol. 6, no. 3, pp. 238–242, 2020. <https://doi.org/10.1016/j.ict.2020.07.002>
- [32] L. Sani, G. Lombardo, R. Pecori, P. Fornacciari, M. Mordonini, and S. Cagnoni, "Social Relevance Index for Studying Communities in a Facebook Group of Patients," in *Applications of Evolutionary Computation (EvoApplications 2018)*, ser. Lecture Notes in Computer Science, vol. 10784, Springer, Cham, 2018, pp. 125–140. https://doi.org/10.1007/978-3-319-77538-8_10
- [33] G. Lombardo, G. Trimigno, M. Pellegrino, and S. Cagnoni, "Language Models Fine-Tuning for Automatic Format Reconstruction of SEC Financial Filings," *IEEE Access*, vol. 12, pp. 31249–31261, 2024. <https://doi.org/10.1109/ACCESS.2024.3370444>
- [34] M. J. J. Bucher and M. Martini, "Fine-Tuned 'Small' LLMs (Still) Significantly Outperform Zero-Shot Generative AI Models in Text Classification," arXiv:2406.08660, 2024. <https://arxiv.org/abs/2406.08660>
- [35] D. Ashok and Z. C. Lipton, "PromptNER: Prompting for Named Entity Recognition," arXiv:2305.15444, 2023. <https://arxiv.org/abs/2305.15444>
- [36] M. V. Muntean, "Multi-Agent System for Intelligent Urban Traffic Management Using Wireless Sensor Networks Data," *Sensors*, vol. 22, no. 1, p. 208, 2022. <https://doi.org/10.3390/s22010208>
- [37] F. León, "ActressMAS, a .NET Multi-Agent Framework Inspired by the Actor Model," *Mathematics*, vol. 10, no. 3, p. 382, 2022. <https://doi.org/10.3390/math10030382>
- [38] F. Balbo, R. Mandiau, and M. Zargayouna, "Extended Review of Multi-Agent Solutions to Advanced Public Transportation Systems Challenges," *Public Transport*, vol. 16, pp. 159–186, 2024. <https://doi.org/10.1007/s12469-023-00332-9>
- [39] S. Huang et al., "Urban Spatial Epidemic Simulation Model: A Case Study of the Second COVID-19 Outbreak in Beijing, China," *Transactions in GIS*, vol. 26, no. 1, pp. 493–511, 2022. <https://doi.org/10.1111/tgis.12850>
- [40] G. White, A. Zink, L. Codecà, and S. Clarke, "A Digital Twin Smart City for Citizen Feedback," *Cities*, vol. 110, p. 103064, 2021. <https://doi.org/10.1016/j.cities.2020.103064>
- [41] B. Min et al., "Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023. <https://doi.org/10.1145/3605943>
- [42] M. Mumtarin, M. S. Chowdhury, and J. Wood, "Large Language Models in Analyzing Crash Narratives – A Comparative Study of ChatGPT, BARD and GPT-4," arXiv:2308.13563, 2023. <https://arxiv.org/abs/2308.13563>
- [43] D. Li et al., "Large Language Models for Generative Information Extraction: A Survey," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186357, 2024. <https://doi.org/10.1007/s11704-024-40555-y>