University *of Ljubljana*
Faculty *of Computer and Information Science*

Automatic generation of Slovenian traffic news for RTV Slovenija

Aljaž Justin, Edin Ćehić and Lea Briški

Abstract

This project addresses the task of automated generation of consistent and factually accurate Slovenian traffic reports for RTV Slovenija using Large Language Models (LLMs). Key methodologies included a data pre-processing pipeline for aligning raw input data with the target reports using Slovene BERT embeddings. Prompt engineering strategies with various LLMs were explored. Subsequently, the Slovenian GaMS model was fine-tuned via Low-Rank Adaptation (LoRA) on an HPC cluster. Its performance was evaluated using quantitative metrics (BLEU, ROUGE, METEOR) and LLM-based quality assessments. Results for configuration with a temperature of 0.4 and repetition penalty of 1.2 yielded optimal performance, achieving strong lexical similarity (ROUGE-L F1 of 0.6) and high scores for factual accuracy and qualitative aspects, showing that this is a viable approach for automatic traffic report generation.

Keywords
LLM, Traffic report, Traffic, Report generation

Advisors: Slavko Žitnik

Introduction

Providing timely and accurate traffic information is a vital public service. This project explores the challenge of automating the generation of factually correct and stylistically suitable Slovenian radio traffic reports for RTV Slovenija. Traditional report generation can be time-consuming, out-of-date and prone to human mistakes. Leveraging Large Language Models (LLMs) offers a way to optimize the process.

The first step was data pre-processing and mapping the two main data sources, tabular traffic data and the structured target reports, by first extracting traffic events and embedding them for semantic matching. Prompt engineering strategies using pre-trained LLMs were explored. Our experimental focus, was on fine-tuning of the Slovenian GaMS model using Low-Rank Adaptation (LoRA) on a supercomputer using a concise prompt, followed by post-processing. The fine-tuned model was evaluated for various configurations by assessing quantitative NLP metrics (BLEU, ROUGE, METEOR) and the bigger LLM model's evaluation of qualitative aspects.

This report presents the methodologies and evaluation findings and outlines potential future work, thus contributing to the development of automated traffic news generation in Slovenian.

Related works

This work focuses on the generation of traffic reports from the raw traffic data. This intersects with several areas of research, including data-to-text generation, NLP for traffic data, and the application of LLMs.

Data-to-Text Generation: The task of generating descriptions from structured data has been a long-standing research area [1]. Our work contributes to this field by focusing on the domain of traffic data. Recent advancements in LLMs have significantly impacted data-to-text generation, by generating more natural and relevant outputs [2].

Traffic Information Processing with NLP: Several works have explored the use of NLP for processing information from traffic sources. For instance, [3] investigates real-time reporting systems using NLP-processed social media data. Our work complements this by focusing on generating reports directly from structured traffic data. The automation of traffic incident management has also been explored recently [4].

Large Language Models for Text Generation and Understanding: The rise of LLMs has opened new avenues for various NLP tasks, including text generation [5, 6, 7, 8]. Techniques such as prompt engineering [9] and fine-tuning [10, 10] are crucial for adapting models to specific domains and tasks. The ability of LLMs to use document layout has

been researched [11]. Article **A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT** explores formatting prompts for LLMs for ensuring form and rules of the output, which could be used for prompt engineering for traffic report generation.

Methods

Our approach started with exploration and pre-processing of data, which was crucial for understanding the problem domain and creating the dataset. Various LLMs and prompt engineering techniques were explored. Furthermore, Slovenian LLM was fine-tuned for the specific task and evaluated using quantitative and qualitative metrics.

Initial Model Exploration and Prompt Engineering

Initially, the focus was on leveraging larger pre-trained LLMs, such as Google's Gemini Flash 2.0 via API, due to its large parameter size and context window, for report generation. Then, prompt engineering experiments were conducted using smaller models, such as Mistral-AI (7B), Llama 2 (17B), Llama 3 (8B), and DeepSeek (7B).

It was observed that the performance of smaller models when using complex prompts with multiple examples and instructions was inferior to that of larger models. Simpler template-based prompts yielded better, but still suboptimal, results. The lack of proficiency in Slovenian often resulted in grammatical errors and unnatural phrasing, sometimes even generating English text. Strict stylistic requirements were difficult to achieve consistently, which was improved if two-step prompt engineering was used, first for report generation and then for improvement of the output quality.

These findings showed the limitations of prompt engineering with available general-purpose smaller models for our goal. While larger models performed well, the need for a more adaptable and more resource-efficient solution led us to focus on fine-tuning of a Slovene-specific model, GaMS-9B.

Data Pre-processing

A significant challenge was the nature of the provided data. We were supplied with two main data sources: 1) tabular time-stamped raw traffic event data, often consisting of many individual entries from DARS, and 2) traffic reports as broadcasted by RTVSlo. A direct one-to-one mapping between individual raw input lines and the final compiled reports was not apparent, as human reporters aggregate, filter, and rephrase information. To create a structured dataset, we implemented a multi-step pre-processing pipeline.

Step 1: Event Extraction from RTV Reports

To understand the events covered in each RTVSlo report, we instructed Gemini 2.0 to extract individual traffic events as a structured string: `LOKACIJA: <lokacija> | SMER: <smer> | TIP: <tip dogodka>`. However, this approach was too slow due to the API limitations, so the reports were heuristically divided into meaningful units instead. Each line of the report, separated by a blank line, usually related to one or connected traffic events.

Step 2: Aligning Extracted Events with Raw Input Data

Once we had a list of specific events for each report, the next step was to find the most relevant entries in the input data that corresponded to these extracted events. The alignment process involved matching based on the semantic similarity using sentence embedding using Slovene BERT-based model, `rokn/slovlo-v1` for events from Step 1) and the raw input segments. The information contained data from columns that represented different traffic information categories (e.g., `ContentNesreceSLO`, `ContentZastojiSLO`). Certain field (like `A1`, `B1`) were furthermore divided. Column names were sometimes prepended as context (e.g., "Nesrece: event description").

For each RTVSlo report and its extracted events, the rows of raw input entries within the temporal window before the report's timestamp were selected (entries from the last 1.5 hours, looking at the last 30-50 records). For each extracted event, its BERT embedding was compared against the embeddings of all text segments within the selected window of input data using cosine similarity. The raw input segments with the highest similarity scores were considered the most relevant sources for that specific event. If the similarity was below a threshold, the input data was not good enough and the report was excluded from the pipeline.

The collection of these best-matching segments for all events of the report was aggregated and paired with the original RTV report to form an input-output pair in the dataset. Cleaning was performed on the target RTV reports mainly to remove the header lines, so that the model would learn to generate only the core content.

Fine-tuning GaMS-9B for Traffic Report Generation

We fine-tuned the GaMS-9B-Instruct model for the task, by data preparation, configuration of the fine-tuning parameters using LoRA, and execution on the HPC cluster.

Model and Environment

The base model selected for fine-tuning was **GaMS-9B-Instruct**. Fine-tuning was performed using PyTorch and the Hugging Face Transformers library. To manage computational resources efficiently, Parameter-Efficient Fine-Tuning (PEFT) was employed, specifically using LoRA. The fine-tuning jobs were executed on an HPC cluster with each job typically requesting 2 GPUs, 16 CPUs per task, and 160GB of memory, with a time limit of 24 hours.

Data Preparation and Loading

The dataset for fine-tuning contained pairs of preprocessed input traffic data and corresponding target RTV reports.

Initial prompts were highly detailed. Their length and complexity caused training failures during GaMS fine-tuning. Therefore, a concise Slovenian prompt with main requirements and the input data was prepared to fit within token limits. The simplified prompt can be found in the appendix.

The expectation was that the desired stylistic and content selection nuances would be learned through the fine-tuning process itself through patterns in the provided input-output.

Each training instance was formatted into a conversational structure: the instructions and pre-cleaned related traffic input formed the 'user', while the target report formed the 'model' turn. The entirety was wrapped with `<bos>...<eos>` tokens. The Hugging Face `AutoTokenizer` was used. A pad token was set to the EOS token if not already present. Inputs were tokenized, padded to `MAX_SEQ_LENGTH` (512 tokens), and truncated if necessary. To ensure the model only learnt to predict the target report, the tokens corresponding to the user message (instruction and input data) in the 'labels' were masked by setting them to -100.

Fine-tuning with LoRA

To efficiently adapt the model for the specific task, the model was prepared for k-bit training, then the LoRA configuration was applied using `get_peft_model`.

For each of the 3 epochs, training loop involved calculation of the loss, performing backpropagation, and updating model weights using AdamW optimizer with a learning rate of 2e-4 and a linear learning rate scheduler. We used a batch size of 2, augmented by 8 gradient accumulation steps, which led to an effective batch size of 16. Model checkpoints and training logs, including average loss per epoch, were saved.

Generation and Testing

The performance was assessed after each training epoch and through comprehensive evaluations executed via SLURM. The fine-tuned weights were loaded to the base model from a checkpoint. The conversational prompt structure and the user message mirrored those of the training phase for consistency. Generation parameters - `max_new_tokens`, `temperature`, and `repetition_penalty`, were configured.

The rule-based post-processing addressed grammatical and stylistic issues of the output, such as converting passive to active voice where appropriate, standardizing terminology, ensuring proper punctuation and spacing. The final report included a standard header with timestamp.

Evaluation

To assess the performance of the fine-tuned model and to understand the impact of different parameters, a combination of quantitative metrics and LLM-based qualitative metrics evaluation was used.

Evaluation Methodology

Five different generation configurations (varying temperature and repetition penalty, identified by the suffix output file (e.g., 57975885)) were evaluated on the test dataset.

Quantitative Metrics

Standard NLP metrics were calculated to compare the generated and the ground truth (target) reports. **BLEU (Bilingual Evaluation Understudy)** measures n-gram precision between the candidate and reference. **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** calculates overlap of n-grams, word sequences, and word pairs. **METEOR (Metric for Evaluation of Translation with Explicit Ordering)** is based on unigram precision and recall, with a penalty for incorrect word order, considering stemming and synonymy.

LLM-based Heuristic Qualitative Evaluation

Gemini was prompted to assess qualitative aspects of the generated report on a scale of 0 to 10 with a brief justification.

Factual Correctness	
1.a	Percentage of correctly identified locations
1.b	Percentage of correctly identified event types (e.g., accident, jam)
1.c	Percentage of correctly identified directions
1.d	Number of events in the generated report not present in the target (scored as $10 - \text{count}$)
Qualitative Aspects	
2.a	Conciseness
2.b	Grammatical correctness
2.c	Stylistic appropriateness

Table 1. Qualitative LLM-based evaluation metrics

Results and Analysis

Five configurations of generation parameters were tested and evaluated using quantitative and LLM-based metrics.

Impact of Generation Parameters on Quantitative Metrics

Table 2 summarizes the average BLEU, ROUGE F1-scores, and METEOR scores for each configuration.

Analysis of Quantitative Metrics: As observed in Table 2, lower temperatures (0.3-0.4) and lower repetition penalties (1.1-1.2) generally yielded higher scores across all quantitative metrics. Configuration C2 shows slightly better ROUGE and METEOR scores compared to C1, which has a slightly better BLEU score. As temperature and repetition penalty increase (C3 to C5), there is a clear and significant degradation in performance across all these lexical similarity metrics. This suggests that more deterministic generation (lower temperature) and less aggressive penalization of repetition lead to outputs that are lexically closer to the target reports.

Impact of Generation Parameters on LLM-based Heuristic Evaluation

Table 3 presents the average scores (0-10 scale) from the Gemini 2.0 Flash heuristic evaluation for each configuration.

Analysis of LLM-based Heuristic Evaluation: The Gemini evaluations, summarized in Table 3, largely corroborate the findings from the quantitative metrics. Configuration C2 (Temp=0.4, RP=1.2) achieved the highest average scores for both factual correctness (7.53) and qualitative aspects (8.71). Similar to the lexical metrics, as temperature and repetition penalty increase (C3 to C5), there is a marked decline in the quality of generated reports. Specifically, higher temperature and repetition penalty led to reports that were less

Table 2. Average Quantitative Metrics for Different Generation Configurations

Metric	**C1 (T=0.3, RP=1.1)**	**C2 (T=0.4, RP=1.2)**	**C3 (T=0.6, RP=1.4)**	**C4 (T=0.8, RP=1.6)**	**C5 (T=1.0, RP=1.8)**
File ID Suffix	...5885	...6261	...6312	...6442	...6558
BLEU	0.377	0.363	0.241	0.096	0.090
ROUGE-1 F1	0.659	0.670	0.544	0.231	0.216
ROUGE-2 F1	0.476	0.482	0.334	0.132	0.123
ROUGE-L F1	0.579	0.593	0.439	0.199	0.185
METEOR	0.644	0.655	0.531	0.262	0.244

C = Configuration, T = Temperature, RP = Repetition Penalty.

Table 3. Average LLM-based Heuristic Evaluation Scores for Different Generation Configurations

Gemini Metric	**C1 (T=0.3, RP=1.1)**	**C2 (T=0.4, RP=1.2)**	**C3 (T=0.6, RP=1.4)**	**C4 (T=0.8, RP=1.6)**	**C5 (T=1.0, RP=1.8)**
File ID Suffix	...5885	...6261	...6312	...6442	...6558
1a. Correct Location (%)	6.80	7.15	5.40	3.45	3.70
1b. Correct Event Type (%)	8.35	8.80	7.20	4.50	5.45
1c. Correct Direction (%)	8.20	8.05	6.05	2.70	3.30
1d. Absence of Extra Events	5.05	6.10	4.60	1.70	1.75
Avg. Factual Correctness	**7.10**	**7.53**	**5.81**	**3.09**	**3.55**
2a. Conciseness	7.80	8.20	7.50	2.30	2.45
2b. Grammatical Correctness	9.90	10.00	10.00	3.55	4.30
2c. Stylistic Appropriateness	8.20	8.85	8.30	2.65	3.15
2d. Overall Quality	7.00	7.70	5.95	2.45	2.70
2e. Radio Suitability	8.00	8.50	6.55	2.40	2.85
Avg. Qualitative Aspects	**8.18**	**8.71**	**7.66**	**2.67**	**3.09**

C = Configuration, T = Temperature, RP = Repetition Penalty

factually accurate (e.g., incorrect locations, types, directions, and more extraneous or hallucinated events) and scored lower on conciseness, style, overall quality, and radio suitability. Grammatical correctness remained high for C1, C2 and C3, but dropped significantly for C4 and C5, indicating that very high temperature/repetition penalty can lead to incoherent or grammatically flawed outputs.

The LLM evaluation provides more nuanced insights. For instance, while C1 and C2 have comparable quantitative scores, C2 is rated slightly higher on qualitative aspects like conciseness and stylistic appropriateness. The justifications often pointed to issues like hallucinated details or misinterpretation of input events in the lower-performing configurations (C4, C5), which lexical metrics alone might not fully capture. For example, reports from C4 and C5 were often flagged as containing completely irrelevant information or being stylistically inappropriate.

This shows the importance of parameter selection. Lower temperatures provide more factual and lexically similar outputs, but lack diversity. However, for this task factual accuracy and specific style are crucial, making lower temperature and moderate repetition penalty better. Postprocessing was used to standardize the format and correct minor, recurrent errors.

Conclusion of Evaluation

The evaluation indicates that the fine-tuned model, particularly where the generation parameters were set to a temperature of 0.4 and a repetition penalty of 1.2 (configuration C2), produces the most balanced results, achieving good scores on both quantitative and qualitative evaluations. Increasing temperature and repetition penalty further led to a decline in report quality, with higher configurations often producing irrelevant, incorrect, or stylistically poor outputs.

Conclusion

This project successfully demonstrated the feasibility of automating Slovenian traffic news generation for RTVSlo by leveraging LLMs. We developed a pipeline involving sophisticated data preprocessing to align traffic data, explored prompt engineering strategies with various LLMs and successfully fine-tunned the Slovenian GaMS-9B model using LoRA on a HPC cluster, which, particularly with a temperature of 0.4 and repetition penalty of 1.2, showed promising results in generating coherent and stylistically appropriate traffic reports based on quantitative and qualitative metrics. The post-processing ensured the quality and consistency of the final outputs.

While the initial objective was achieved, exploring advanced fine-tuning techniques (e.g., different PEFT methods, full fine-tuning), optimizing parameters, refining outputs with post-processing or learned correction models, might lead to further improvement. Deeper domain-specific understanding could be achieved via dataset enrichment or advanced prompting. In the future, translation and Text-to-Speech (TTS) could further enhance the generation. Before employing the automatic generation system for RTVSlo, comprehensive human evaluation and error analysis should be performed.

In conclusion, the developed methods for data processing, prompt engineering, fine-tuning, and evaluation provide a path for development of an efficient and quality system for automatic traffic report generation.

References

[1] Microsoft Research. Data2text: Automated text generation from structured data, 2025. Accessed: 2025-03-19.

[2] LlamaIndex. Building blocks of llm report generation: Beyond basic rag, 2025. Accessed: 2025-03-19.

[3] Xiangpeng Wan, Michael Lucic, Hakim Ghazzai, and Yehia Massoud. Empowering real-time traffic reporting systems with nlp-processed social media data. *IEEE Open Journal of Intelligent Transportation Systems*, 1:159–175, 01 2020.

[4] Matteo Cercola, Nicola Gatti, Pedro Huertas Leyva, Benedetto Carambia, and Simone Formentin. Automating the loop in traffic incident management on highway, 2025.

[5] Domen Vreš, Martin Božič, Aljaž Potočnik, Tomaž Martinčič, and Marko Robnik-Šikonja. Generative model for less-resourced language with 1 billion parameters, 2024.

[6] Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. Multilingual machine translation with large language models: Empirical results and analysis, 2024.

[7] Elijah Pelofske, Vincent Urias, and Lorie M. Liebrock. Automated multi-language to english machine translation using generative pre-trained transformers, 2024.

[8] Xirui Peng, Qiming Xu, Zheng Feng, Haopeng Zhao, Lianghao Tan, Yan Zhou, Zecheng Zhang, Chenwei Gong, and Yingqiao Zheng. Automatic news generation and fact-checking system based on language processing, 2024.

[9] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt, 2023.

[10] Mathav Raj J, Kushala VM, Harikrishna Warrier, and Yogesh Gupta. Fine tuning llm for enterprise: Practical guidelines and recommendations, 2024.

[11] Marcel Lamott, Yves-Noel Weweler, Adrian Ulges, Faisal Shafait, Dirk Krechel, and Darko Obradovic. Lapdoc: Layout-aware prompting for documents. In Elisa H. Barney Smith, Marcus Liwicki, and Liangrui Peng, editors, *Document Analysis and Recognition - ICDAR 2024*, pages 142–159, Cham, 2024. Springer Nature Switzerland.

Conscise Prompt for Fine-tuning

```
Generiraj prometno poročilo v
slovenščini za radio.
Upoštevaj standardno strukturo:
1. Začni z "Prometne informacije
[datum] [čas] za Radio Slovenija"
2. Vključi samo
pomembne prometne dogodke
3. Uporabljaj
trpne oblike in ustrezno terminologijo
4. Poročilo naj
bo jedrnato (<1 min branja)

Podatki o prometu: <>
```

Qualitative Examples

Examining specific examples from the output files *57976261* for C2 vs. *57976558* for C5 illustrates these differences.

For instance, for the input events on *2022-01-10 17:30:00*:

- **Input Events:** ”Cesta čez prelaz Vršič je zaprta. Na gorenjski avtocesti pred predorom Karavanke proti Avstriji zastoj tovornih vozil, 500 m. Čakalna doba je na mejnih prehodih Slovenska vas, Obrežje in Gruškovje.”
- **Target Report (excerpt):** ”Cesta čez prelaz Vršič je zaradi zimskih razmer zaprta. Pred predorom Karavanke je približno kilometer in pol dolg zastoj tovornih vozil proti Avstriji. Na mejnem prehodu Obrežje vozniki osebnih vozil na vstop v Slovenijo čakajo 2 uri, v Gruškovju in Slovenski vasi pa pol ure...”
- **Generated C2 (T=0.4, RP=1.2) (excerpt):** ”Cesta čez prelaz Vršič je zaradi zimskih razmer zaprta. Pred predorom Karavanke je kilometer dolg zastoj tovornih vozil proti Avstriji. Na mejnem prehodu Obrežje vozniki osebnih vozil na vstop v državo čakajo do dve uri, v Gruškovju pa pol ure...” This report is largely accurate and stylistically appropriate.
- **Generated C5 (T=1.0, RP=1.8) (excerpt):** ”Cesta čez Prelazovma Korensko sedlo in Ljubelj sta zaradi zimskih pogojev še vedno zaprti; cesta cez azVršič pa bo do nadaljnjega ostala odprta samo po cestišču z verigami ali snegom polnimi pnevmatikah.. Predvidoma jutri popoldan bodo začeli s puljenjem prometa skozi PredoreKaravanké vozeča se trajala približno dve uri...” This output is significantly degraded, containing irrelevant information, misspellings, and non-Slovenian characters, making it unsuitable.