



# Automatic Generation of Slovenian Traffic News for RTV Slovenija

Filip Turk and Tschimy Aliage Obenga

## Abstract

The abstract goes here.

## Keywords

Generating traffic reports, Large Language Models, NLP, Prompt Engineering, Fine-tuning, Slovenian traffic news, Automated text generation

Advisors: Slavko Žitnik

## Introduction

Traffic reporting is a crucial aspect of public broadcasting, especially for real-time updates on road conditions. Currently, RTV Slovenija relies on students to manually check, filter, and type reports from the Promet.si portal every 30 minutes. This process is time-consuming and prone to inconsistencies.

This project aims to automate traffic news generation using a Large Language Model (LLM). The approach includes leveraging prompt engineering techniques, defining evaluation criteria, and fine-tuning an LLM to improve accuracy and relevance. The generated reports must align with RTV Slovenija's guidelines, ensuring clarity, conciseness, and correctness in road naming and event significance.

## Existing Solutions and Related Work

### Traffic News Automations

Automated traffic reporting systems have been developed worldwide, primarily using Natural Language Processing (NLP) and Machine Learning (ML) techniques. Some existing solutions include:

1. Google Maps Traffic Alerts: Uses real-time data and machine learning to generate concise traffic updates
2. Waze Traffic Reports: Crowdsourced traffic conditions, automatically summarized for users.
3. AI-driven News Generation: Organizations like OpenAI and Google have explored AI-generated news reports in sports, finance, and weather domains.

These solutions utilize structured traffic data and LLMs for automated summarization. However, fine-tuning on local

datasets, such as Slovenian road networks and traffic terminology, remains a challenge.

### Large Language Models in Text Generation

Recent advancements in LLMs like GPT-4, LLaMA, and T5 have demonstrated strong capabilities in text summarization and structured data interpretation. Studies have shown that fine-tuning domain-specific models improves text coherence and factual accuracy.

## Initial Corpus Analysis

The dataset for this project consists of structured traffic data from Promet.si, formatted in Excel. A preliminary analysis reveals the following key attributes:

Road Names and Locations: Entries specify highways, regional roads, and key urban intersections.

Incident Types: Traffic congestion, road closures, weather-related disruptions, and accidents.

Time Stamps: Time of report updates, crucial for real-time relevance.

Severity and Impact: Differentiates minor delays from major traffic disruptions.

Initial observations suggest that filtering out redundant or low-impact reports will be necessary to maintain concise news summaries. To achieve this, categorizing incidents based on importance will help prioritize critical updates. Categories could include:

1. High Priority: Major accidents, full road closures, severe weather disruptions

2. Medium Priority: Traffic congestion, partial road closures, significant delays
3. Low Priority: Minor roadworks, temporary disruptions with minimal impact

This structured approach will enhance the efficiency of traffic news generation by ensuring that significant events receive appropriate attention.

## Initial Ideas and Approach

### Prompt Engineering

The first step involves experimenting with LLMs through structured prompts to generate effective traffic summaries. Potential techniques include:

Using role-based prompting (e.g., “You are a Slovenian traffic news reporter”)

Applying structured templates to ensure consistent output

Fine-tuning prompt phrasing to optimize summarization quality

### Evaluation Criteria Definition

To ensure quality traffic reports, evaluation metrics will focus on:

1. Accuracy: Correct road names and locations
2. Conciseness: Keeping reports within an optimal word count
3. Format: Ensuring the report fit the provided format

### Parameter-Efficient Fine-Tuning

Once prompt engineering is refined, fine-tuning an LLM (such as GPT-3.5 or T5) with a domain-specific dataset will be explored. LoRA (Low-Rank Adaptation) or Adapter-based methods will be considered to optimize training efficiency.

### Interactive Testing Interface

A web-based or command-line interface will be developed to allow interactive testing of generated reports before full automation.

## Work Done So Far

### Data Collection and Preprocessing

We processed and paired input traffic report data with their corresponding output RTF files (traffic announcement files), ensuring a one-to-one mapping between inputs and announcements. This paired dataset serves as our training corpus for fine-tuning the language model. The data underwent preprocessing to create a structured format where each example consists of a user message (input traffic report) and an assistant message (official traffic announcement).

### Model Selection and Setup

For the initial language model, we selected `cjvt/GaMS-1B`, which is suitable for the Slovenian language and efficient for fine-tuning. The model was loaded using 4-bit quantization (`BitsAndBytes`) to enable efficient training and inference on limited hardware (Google Colab with T4 GPU). The tokenizer was configured to ensure compatibility with the model and Slovenian text.

### Prompt Engineering

We designed few-shot prompts with multiple diverse examples, covering accidents, traffic jams, weather-related delays, and road closures. Explicit instructions were included in the prompt to enforce RTV Slovenia’s reporting style, road naming conventions, and prioritization (A1, B1, C1). Both raw text and structured input formats were tested to improve model consistency.

### Fine-Tuning

Low-Rank Adaptation (LoRA) was used for efficient fine-tuning of the model on the traffic dataset, significantly reducing the number of trainable parameters. The `trl` library’s `SFTTrainer` was used with custom training arguments. The dataset was converted into a format suitable for supervised fine-tuning, with clear input-output pairs.

### Evaluation

At this stage, formal evaluation metrics have not yet been implemented. Initial testing has been conducted through manual verification, where we generated reports for random test inputs and compared them to expected outputs. This preliminary testing helped us identify critical issues in our approach and guided our model selection process.

Our initial experiments with `TinyLlama/TinyLlama-1.1B-Chat` showed limited success, primarily due to its challenges with Slovenian language generation and domain-specific content. This led us to switch to `cjvt/GaMS-1B`, which demonstrated significantly better performance in generating coherent and contextually appropriate traffic announcements in Slovenian.

While these initial results are encouraging, we recognize the need for:

- Implementation of quantitative evaluation metrics (BLEU, ROUGE)
- Structured evaluation using our Pydantic schema
- Systematic testing across different types of traffic events
- Assessment of factual accuracy and consistency

## Analysis of Results

### Strengths

- The model generates structured traffic reports that closely follow the required format and style.

- 4-bit quantization and LoRA allow for efficient training and inference on modest hardware.
- With diverse few-shot examples, the model can handle various event types (accidents, weather, jams, etc.).

### Weaknesses

- The model sometimes repeats the same output multiple times.
- Overfitting to frequent examples can occur, making it difficult to generalize to rare or novel inputs.
- Raw, unstructured input can confuse the model; structured input formats improve consistency.

### Lessons Learned

- Clear, structured prompts with diverse examples yield better results.
- Using a schema clarifies the task for both the model and the developer.
- LoRA fine-tuning is practical for domain adaptation with limited resources.

## Next Steps

### Model Enhancements

- **Structured Input/Output:** Move towards using the schema directly as input/output for the model, possibly with serialization (e.g., JSON).
- **Error Analysis:** Systematically evaluate model errors and refine prompts or training data accordingly.
- **Evaluation Metrics:** Implement quantitative evaluation (BLEU, ROUGE, or custom metrics for factual accuracy).
- **Model Variants:** Use different models and compare results.

### Deployment

- **API Integration:** Wrap the model in an API for real-time traffic report generation.
- **User Interface:** Develop a simple web or mobile interface for end-users (e.g., journalists or the public).
- **Continuous Learning:** Set up a feedback loop to incorporate corrections and new data into future fine-tuning rounds.

## Conclusion

Significant progress has been made in automating traffic report generation for RTV Slovenia using large language models and LoRA fine-tuning. The system can already produce structured, high-quality reports, but further improvements in data, model robustness, and evaluation are planned. The next phase will focus on scaling, deployment, and continuous improvement.

## References

- [1] Wolf, T., et al. (2020). *Transformers: State-of-the-Art Natural Language Processing*. HuggingFace. <https://huggingface.co/docs>
- [2] Hu, E. J., et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv preprint arXiv:2106.09685. <https://arxiv.org/abs/2106.09685>
- [3] von Werra, L., et al. (2023). *TRL: Transformer Reinforcement Learning*. HuggingFace. <https://huggingface.co/docs/trl>
- [4] Ulčar, M., et al. (2023). *GaMS: Generative AI Models for Slovenian*. CJVT. <https://huggingface.co/cjvt/gams-1b>
- [5] Lhoest, Q., et al. (2021). *Datasets: A Community Library for Natural Language Processing*. HuggingFace. <https://huggingface.co/docs/datasets>
- [6] RTV Slovenija. (2024). *Traffic Report Guidelines and Standards*. Internal documentation.