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

Automatic generation of Slovenian traffic news for RTV Slovenija

Janez Tomšič, Žan Pušenjak, Matic Zadobovšek

Abstract

In this report we present our initial data preparation process on the dataset of RTV Slovenija traffic news. We outline the language models we are planning to use for generating reports using prompt engineering and fine-tuning techniques. Lastly, we outline the evaluation procedure we will use, which consists of automated metrics (BLEU, BERTscore), and manual human evaluation.

Keywords
Traffic news generation, Prompt engineering, Supervised fine-tuning, Natural language generation

Advisor: Slavko Žitnik

Introduction

Generating traffic news using the modern Natural Language Processing (NLP) techniques involves transforming raw traffic data into structured reports, that are in a form that is suitable to be broadcast on radio. Our main goal is to reduce manual workload for editors, and to improve the quality and consistency in prepared reports.

There are a few challenges that arise, mainly in following the formatting rules and correctly reflecting traffic conditions (e.g. road names, directions) - this is what the model has to learn to replicate. To address these conditions, we will use pre-trained language models, structured prompting, and fine-tuning. We combine all these techniques together in order to be able to automatically generate high-quality Slovenian traffic reports.

Related Work

We looked at the popular approaches for text style generation adaptations for language models. The two most prominent techniques are *prompting* and *fine-tuning*. Both try to achieve better desired results either by guiding the model to generate data according to determined rules or to change the models structure in accordance with the desired outcome.

The available research suggests that *prompting* is overall less efficient than *fine-tuning* [1] but should not be discarded. Using the right prompt can substantially increase the models desired performance [2]. Additionally smaller and medium-sized models benefit more from prompting than from simply increasing the model size [1]. The prompt should be crafted according to the desired result [3], so it is important to not rush the process of finding the best performing one. Taking that into account we conducted that crafting a good prompt for the style of the output is essential for a good generation of the output data.

If *prompting* is enhanced by using a *fine-tuned* model, we can achieve the best possible results. For our use case we would use *fine-tuning* to make the model adapt the generated content style to the form that is used on RTV. The problem with *fine-tuning* is lack of tuning data and computing power limitations [4]. But apart from those problems the approach can heavily change and customize the generation style with noticeably better results [5].

Attempts at creating an automated traffic incident reporting were already made [6] so looking at the findings and shortcomings of existing systems will allow us to even further improve the results. Since the traffic data and the desired report structure vary greatly even between news channels, let alone countries, it is impossible to create an out-of-the box solution for news reporting.

Traffic News Dataset

In this section we present the provided dataset and the preprocessing steps we applied to the dataset to make it more suitable for language models. The dataset consists of two parts. The first part is a table that was retrieved from the Slovenian traffic website **promet.si** [7] which provides real-time traffic updates and road conditions. The second part is

a corpus of traffic reports that were manually written to be read on the radio broadcast. Both parts span data from 2022 to 2024. Additionally, a set of instructions which outline the expected format of generated traffic news reports are included in the dataset to help with consistency.

Tabular Traffic Updates Data

Traffic updates from promet.si are stored in a structured Excel table. It contains a separate columns for different categories, such as accidents, traffic jams, roadworks, and obstacles. There are also two columns where this data is aggregated, one containing the urgent traffic updates and one containing general traffic updates.

Not much preprocessing was required on this table, as the data is already structured. Nonetheless, two important preprocessing steps had to be applied. Firstly, the entries were *converted from HTML* to a raw textual representation which is more suitable for LLMs. Secondly, we found that the timestamps in the table were given in the *UTC timezone*. We localized these timestamps to make the table compatible with the unstructured reports described in the next section.

Corpus with Traffic Reports

The corpus contains 28,037 manually written traffic reports which are stored in rich text format (*rtf*) files. Generally, the first line in the report is a header containing the report type (general or urgent) and a local timestamp. The main report content is contained in the remaining lines. Only the reports for which the header was successfully parsed were kept, as obtaining the timestamp is crucial for matching the reports with the tabular data.

After parsing the reports, empty reports were removed from the dataset. Furthermore, outliers were identified by embedding each report using a SloBERTa model, fine-tuned for topic classification [8]. The average cosine similarity between the embedding of each report and the other report embeddings was computed. The reports with the lowest average similarities were manually inspected. As a result, one additional report was excluded from the dataset, since it was not a traffic report.

Matching the Reports with Tabular Data

As a final data preparation step, we matched the parsed reports with the entries in the traffic information table. To each report, up to 10 of the most recent entries that occurred up to 30 *minutes* before the report timestamp were assigned. Since some of these entries were identical, duplicated were removed. Note that this configuration is subject to additional changes after further experiments will be conducted.

For some reports, no entries were found in the table in the defined time window of 30 minutes. These reports were excluded from further analysis. An overview of corpus size after each data preprocessing stage is available in Table 1.

Preparation Stage	Number of Reports
Initial	28,037
Parsing Reports	27,567
Identifying Outliers	27,561
Matching with Tabular Data	25,415

Table 1. Size of the corpus after individual data preparation stages.

Report Writing Guidelines

The dataset also contains an auxiliary file with guidelines for writing reports. It specifies that each traffic incident we report on has to follow a structured formulation, starting with a road name and direction, followed by a cause, and lastly by the consequence of the incident. It defines the hierarchy of events that need to be reported.

There are a few other rules and naming conventions that are more specific and are dependent on the type of event we are reporting on (high-priority events must be repeated every 15-20 minutes, congestion is reported only if it exceeds 1 km in length, indicate road sections with expected closures or delays). Note that some of the naming conventions for reports differ from naming conventions on *promet.si*, where the tabular data was retrieved from. This will need to be taken into account in our analysis.

Methodology

In this section, we describe our initial approach on how we could potentially tackle automatic generation of Slovenian traffic news. Our goal is to develop a system that would be able to convert data from initial Excel files into structured reports that follow predefined formatting rules, and are ready for the radio presenters.

Fine-tuning GaMS-2B with LoRA adapters

To adapt the CJVT/GaMS-2B-Instruct model for Slovene traffic reporting, we used a parameter-efficient fine-tuning pipeline on Arnes HPC V100 partition. The key steps were:

Quantized base model We loaded GaMS-2B in 4-bit NF4 precision with double quantization and bfloat16 compute, plus FP32 CPU offload for int8 operations. This reduces GPU memory to fit a single V100.

Data preparation & label masking Our source file contains raw Slovene traffic-event prompts paired with human-written responses. After a 90/10 train/test split, each example is serialized as:

```
### Human: ⟨prompt⟩
### Assistant: ⟨response⟩
```

We tokenize to a fixed length (300 tokens) and construct the `labels` array by assigning -100 (ignore) to all tokens belonging to the human prompt, so that loss is computed only on the output.

LoRA adapter configuration To avoid updating the full 2 B-parameter model, we injected low-rank adaptation modules into the query and value projection matrices of each attention layer:

- **Rank** $r = 4$: small adapter footprint
- $\alpha = 16$: scaling factor for gradient stability
- dropout $= 0.1$: regularization to prevent overfitting

Training hyperparameters We used mixed-precision (fp16) on a single GPU:

- **Batch size**: 1 example per device
- **Epochs**: 3 full passes over the data
- **Learning rate**: 2×10^{-5} with cosine decay and 500 warm-up steps
- **Gradient accumulation**: 1 step (no accumulation)
- **Checkpointing**: save every 1 000 steps, keep last 5

HPC execution The fine-tuning job ran under SLURM on 1 V100 GPU and 4 CPU cores. Total time was approximately 5 hours.

Evaluation

In this section we outline our initial ideas for evaluating the generated traffic reports. We plan to use automated evaluation methods as well as manual human evaluation, especially for evaluating structure of the generated reports, their relevance, and factual accuracy.

For automatic evaluation, we are considering the following methods:

- **ROUGE** [9] is commonly used for evaluating summarization methods. It estimates how much of the reference text is captured by the generated text using either N-grams, skip-grams, or longest common subsequences, depending on the variant used.
- **BERTScore** [10] compares semantic similarity of texts rather than just analysing word overlap. It uses contextual embeddings and cosine similarity. We are considering using *SloBERTa* [11], a RoBERTa-based contextual embedding tuned for the Slovenian language.

The biggest advantage of these methods is that they do not require manual human review. However, they are not able to directly check if the structure of a report is suitable, or whether the reported events in fact occurred. They also can not explicitly check if reporting conventions were taken into account. Additionally, as observed in the data exploration phase, some ground-truth reports do not contain the latest traffic information, which might cause our models to be underestimated.

Because of this, we are also considering manual review of generated reports, at least on a subset of the data. Another idea we have is to attempt to define an automated evaluation method that would be better suited for our task.

When defining the evaluation procedure, additional criteria should also be taken into account. The detection of urgent traffic events should also be evaluated, perhaps some events need to have a higher weight assigned to them - for example, a wrong-way driver on the highway is much more crucial to be reported than a congestion at a border crossing.

References

[1] Martina Toshevska and Sonja Gievska. Llm-based text style transfer: Have we taken a step forward? *IEEE Access*, 13:44707–44721, 2025.

[2] Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. Does prompt formatting have any impact on llm performance?, 2024.

[3] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[4] Yuchen Xia, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, Cong Callie Hao, and Nishil Talati. Understanding the performance and estimating the cost of llm fine-tuning. In *2024 IEEE International Symposium on Workload Characterization (IISWC)*, pages 210–223, 2024.

[5] Xinyue Liu, Harshita Diddee, and Daphne Ippolito. Customizing large language model generation style using parameter-efficient finetuning, 2024.

[6] Anna Jansdatter Bjørgo and Mads Lundegaard. Automating the process of traffic incident reporting. Master's thesis, NTNU, 2024.

[7] DARS d.d. Prometno informacijski center za državne ceste. https://www.promet.si. Accessed: 20. 03. 2025.

[8] CJVT. Sloberta for text topic classification. https://huggingface.co/cjvt/sloberta-trendi-topics. Accessed: 05. 02. 2025.

[9] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[10] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[11] Matej Ulčar and Marko Robnik-Šikonja. Slovenian roberta contextual embeddings model: Sloberta 2.0. *clarin.si*, 2021.