FRI Natural language processing course 2025
Project report

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

Keywords
LLM, Traffic

Advisors: Slavko Žitnik

Introduction

Our initial goal is to leverage an open-source large language model (LLM) capable of generating text in Slovenian to automate the process of traffic report generation for RTV SLO. By doing so, we aim to streamline and enhance the efficiency of producing structured reports while ensuring consistency in formatting and adherence to predefined guidelines.

To achieve this, we plan to train the LLM to learn the specific formatting and stylistic requirements from existing data. This will allow the model to generate reports that align with the required standards without the need for extensive manual editing. Since using a large-scale model can be resource-intensive, we also intend to fine-tune a smaller, more efficient LLM tailored to our specific use case. This smaller model would retain the necessary capabilities while reducing computational costs, making it more practical for real-world deployment.

If time permits, we aim to extend the project by integrating automatic translation features for languages commonly spoken in Slovenia, such as English, German, and Italian. This would make the generated reports more accessible to a wider audience. As a potential final step, we may also explore the implementation of text-to-speech functionality, particularly for English, to further enhance accessibility and usability.

Through this approach, we seek to develop a scalable and efficient solution for automated report generation that meets the needs of Slovenian-speaking users while incorporating multilingual and accessibility features as an added value.

Related works

This work focuses on the automatic generation of traffic reports from traffic data. This intersects with several areas of research, including data-to-text generation, natural language processing for traffic information, and the application of large language models (LLMs).

Data-to-Text Generation: The task of automatically generating textual descriptions from structured data has been a long-standing research area [1]. Our work contributes to this field by focusing on the specific domain of traffic data, aiming to produce informative and concise reports. Recent advancements in LLMs have significantly impacted data-to-text generation, offering new possibilities for creating more natural and contextually relevant outputs [2].

Traffic Information Processing with NLP: Several works have explored the use of NLP techniques for processing and extracting information from traffic-related sources. For instance, [3] investigates empowering real-time traffic 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 using loop data has also been explored recently [4].

Large Language Models for Text Generation and Understanding: The rise of large language models has opened new avenues for various natural language processing tasks, including text generation [5, 6, 7, 8]. Techniques such as prompt engineering [9] and fine-tuning [10, 10] are crucial for adapting these models to specific domains and tasks. Furthermore, the ability of LLMs to understand and utilize document layout for enhanced performance has been investigated [11]. While our work primarily focuses on generating text from structured data, these advancements in LLM capabilities are highly relevant.

Evolution of Collective Behavior and Linguistic Systems: Although seemingly distant, the foundational work on the evolution of collective behavior using linguistic fuzzy rule-based systems [12] and the promotion of parallel movement through balanced antagonistic pressures [13] provide insights into the development of complex communicative behaviors in artificial systems, which can indirectly inform the design of more sophisticated traffic reporting systems in the future.

Article **A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT** explores possibilities of formatting prompts for existing LLMs, guiding them to ensure certain form and rules of the generated output. This could be used for the first part of the task, where our task would lead us to initially use prompt engineering for traffic report generation.

Methods

The first method we employed was **prompt engineering** to generate traffic news based on the data provided in the table. For our initial model selection, we decided to use a larger model — **Gemini Flash 2.0** — due to its ease of use, large parameter size, and extensive context window.

We observed that the HPC environment had multiple issues, including stability problems, frequent downtime, and difficulties compiling necessary libraries. To expedite our testing process, we accessed the Gemini model via an API key, which allowed us to avoid these HPC-related issues.

In addition to API-based testing, we attempted to run models locally via SSH. We successfully loaded the **Mistral-AI model** with 7B parameters and conducted experiments with prompt engineering. We also attempted to load **Llama 4.0 17B with 128 experts**, but we did not initially account for the model's full size (approximately 400B parameters), and therefore were unable to complete testing with it.

As an alternative, we chose to use a smaller version of **Llama 17B** with 16 experts, which should be able to run on a single GPU.

Additionally, for a more competitive smaller model, we selected the **DeepSeek LLM 7B Base** model to further optimize performance and resource usage.

Smaller Models

In regards to lighter, smaller models, Llama version 3 (8B edition) and Deepseek LLM (7B edition) were tested with various prompt approaches. However, any of these approaches were not comparable to the larger models, such as the aforementioned Gemini 2.0. There are multiple approaches to prompting a smaller LLM model. One option was to include the example of a traffic report within the prompt. However, that potentially introduces confusion and mixture of data from which the report should be generated. Providing only a template for the structure of the report improved the issue. Smaller models are not as proficient in the Slovenian language compared to the larger ones. Therefore, to improve the report readability and to keep it understandable and concise, the response to the initial prompt was further processed for grammatical and stylistic improvement. However, without specifying that report should remain in Slovenian, the response was automatically translated into English. Overall, smaller models had more trouble following the stylistic requirements for the traffic reports of this caliber.

Data Preprocessing

We noticed that the data from the report is not generated solely from a single line in the table and that the lines in the table do not correlate one-to-one with the provided news reports. Many details are merged or omitted in the final report, and some information is carried over from previous entries in the traffic data array.

To address these issues, we decided to preprocess our data to enable easier prompt generation and ensure that each record in our training data is clearly labeled. Our goal was to combine multiple lines from the traffic information data into a single, more informative package, thus avoiding repetition in the prompts sent to the model.

For the additional information found in the traffic data tables, we decided to remove columns that were not relevant or were not used in the final reports, such as weather data, general news, and extensive road closure listings. We found that the raw traffic data listed all road closures across the country, but not all of this information was included in the final reports from RTV Slovenija. Therefore, we omitted such details from the prompt preparation as well.

Data Problems

Since the data provided is a real case scenario, there are impurities and problems within the data that can represent an obstacle when generating a traffic report. The intervals of the reports do not follow a strict daily structure. During most hours of the days, there is a 30 minute interval between the traffic reports. However, that is not the case every time. Another anomaly are the cases of emergent traffic reports, such as when there is a driver driving in the opposite direction on a highway, etc. Oftentimes, it is troublesome to determine the interval of the input data that corresponds to the output data. Reaction time, namely the time from the start of the traffic event to when the event was included in the report, is not always even. This is probably due to the human factor of the report generation so far. If the reports were generated solely using LLMs in real time, the reports could be more up to date and reflect the actual traffic situation better. Another problem is deciding whether to include a certain type of input data. Most reports include information about accidents, obstacles and traffic jams as priority information. Sometimes, information about roadwork, weather conditions and general announcements are included as well, but not always, especially if the report is already long enough when it includes other, more important data. Sometimes, reports include information about when the traffic jam is cleared or when there are no more obstacles on the road. Input data rarely includes the information about such events, rather including information about start, not the end of such events.

Prompt Engineering

To help the models generate outputs similar to the desired results, we instructed them to act as RTV Slovenija reporters tasked with creating traffic news reports. We provided detailed guidelines describing how the reports should look.

From analyzing the final reports, we found that the most important aspects the model must be aware of are the need for extremely concise, radio-friendly reports in Slovenian, adherence to language norms, relevance filtering, stylistic consistency, strict length limits, and maintaining an implicit structure without the use of subtitles.

In the initial prompt, we also included a few examples from the dataset that the model could reference to better understand the desired structure and order.

The detailed prompt structure can be found in the file `gemini-api/gemini_API_reporter.py`.

Observations

Our observations from generating reports using prompt engineering show that the models follow the given rules very consistently and are capable of creating reports that are quite similar to those from RTV Slovenija. However, we found that the reports generated by the LLMs are significantly longer, mainly because they include unnecessary details that human reporters typically omit.

Additionally, reporters often simplify the location of accidents or traffic jams to make the information easier for listeners to understand. In contrast, the LLMs tend to be more precise and provide exact locations based on the input data.

Future Directions and Ideas

We plan to test the different LLMs we mentioned earlier in more detail and further optimize the prompting strategies for each model. Due to differences in model size, architecture, and behavior, we expect that prompts will need to be tailored specifically to each model to achieve the best results.

Additionally, we are working on improving our current prompting approach by making the generated reports more concise and by integrating domain-specific knowledge. This should help the models simplify the reporting style and present the locations of traffic incidents in a way that is easier for listeners to understand.

One of our goals is to better align the two provided datasets in order to streamline and simplify the generation of traffic news.

Our final objective is to fine-tune the models on the provided data to encourage the use of language and style that more closely matches that of human reporters. We also aim to help the models internalize domain-specific knowledge by retraining them through the regeneration of traffic reports.

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.

[12] Jure Demšar and Iztok Lebar Bajec. Evolution of Collective Behaviour in an Artificial World Using Linguistic Fuzzy Rule-Based Systems. *PLoS ONE*, 12(1):1–20, 2017.

[13] Jure Demšar, Erik Štrumbelj, and Iztok Lebar Bajec. A Balanced Mixture of Antagonistic Pressures Promotes the Evolution of Parallel Movement. *Scientific Reports*, 6, 2016.