FRI Natural language processing course 2021
Project report

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

Automatic generation of Slovenian traffic news for RTV Slovenija

Katarina Gojković, Alen Kurtagić, and Žan Terplan

Abstract

/

Keywords
Traffic report, LLM

Advisors: Slavko Žitnik

Introduction

Large Language Models (LLMs) have revolutionized natural language processing (NLP) by achieving state-of-the-art performance across a wide range of tasks, including text generation, translation, and summarization. Models like GPT-3 [1] and BERT [2] have transformed the field by demonstrating advanced capabilities in processing and generating human-like text. However, despite their success, LLMs face challenges in less spoken languages with low resources.

Related Work

LLMs have been increasingly used in automated news generation domains, especially in fields like finance, sports, and weather reporting. Despite this, traffic news generation is still a fairly unexplored area. However, some studies have explored this topic, for example, Wan et al. [3] leveraged social media data to complement traditional traffic reporting systems. Ouyang et al. [4] used AI agents to process multi-scale traffic data, conduct analysis and generate suggestions.

Besides traffic news generation, data-to-text generation in general has seen significant advancements with the application of neural network architectures. Several recent studies have explored different methodologies for improving the quality and accuracy of generated text from structured data.

One key advancement in data-to-text generation is the incorporation of entity modeling. Puduppully et al. (2019) [5] proposed an entity-centric neural architecture that dynamically updates entity-specific representations and employs hierarchical attention for generating coherent and factually accurate text. Their approach outperformed existing models on both the benchmark ROTOWIRE dataset and a newly introduced baseball dataset, demonstrating improvements in content selection and text fluency.

Another significant development in the field has been the comparison between pipeline and end-to-end architectures. Castro Ferreira et al. (2019) [6] systematically compared neural pipeline models with end-to-end approaches for generating text from RDF triples. Their findings indicated that pipeline models, which introduce explicit intermediate steps, produce more fluent and accurate text than end-to-end models. Additionally, the pipeline approach was found to generalize better to unseen inputs, reinforcing the notion that structured intermediate representations play a crucial role in high-quality text generation.

Curriculum learning has also been explored to enhance data-to-text generation, particularly in cross-lingual and noisy data settings. Hari et al. (2024) [7] introduced an approach that applies curriculum learning principles by ordering training samples based on an alignment score and using an annealing schedule. Their method resulted in significant improvements in BLEU scores and faithfulness metrics across multiple Indian languages and English.

A recent study by Zhang et al. (2024) [8] proposed optimized LLMs tailored for medical record generation, incorporating data augmentation techniques and domain-specific customization. Their models significantly improved faithfulness scores compared to state-of-the-art general models, demonstrating the importance of high-quality annotated data in generating accurate medical records.

Overall, these studies highlight various approaches to improving data-to-text generation, from entity modeling and pipeline architectures to curriculum learning and domain-specific optimizations.

Table 1. Candidate models

Metric	Architecture	Size	Slovene	Instruct-tuned
GaMS-1B-Chat [9]	Decoder	Medium	Yes	Yes
Mixtral-8x7B-Instruct	Decoder	XL	Yes	Yes
mt0-large	Encoder-Decoder	Medium	Yes	Yes
aya-101	Decoder	XL	Yes	Yes
Llama-3.1-8B	Decoder	XL	Partial	Partial
gemma-2b-it	Decoder	Medium	Partial	Yes
t5-sl-small	Encoder-Decoder	Small	Yes	No

Proposed Approach

Model Selection Criteria

Selecting an appropriate pre-trained language model is a critical step in designing any language generation system. Especially for specialized data-to-text tasks such as generating Slovenian traffic news reports. LLMs differ significantly in terms of architecture, size, multilingual support, and their ability to follow natural language instructions, all of which must be considered.

Firstly, since our input consists of structured data, we consider the encoder-decoder architecture to be the most suitable for this task. Unlike classical text generation, data-to-text generation benefits from models that can explicitly encode the input structure before producing the output. In encoder-decoder models, the encoder processes the input data into a meaningful representation, which the decoder then uses to generate a coherent output. This separation often leads to better consistency and can help reduce hallucinations. That said, we do not rule out the use of decoder-only architectures, especially given their recent success in following instructions.

Since the model will not be deployed on embedded or resource-constrained devices, there is no strict requirement for it to be extremely lightweight. At the same time, the task does not demand highly creative or open-ended text generation, which is where larger models typically excel. Furthermore, the inference time requirements are relatively relaxed, as traffic news reports are generated on a fixed schedule (e.g., every 30 minutes) and not in real-time. Therefore, a mid-sized model that balances performance, generation quality, and reasonable inference time is likely to be the most appropriate choice for this application.

One of the most important factors for this task is the model's ability to generate text in Slovenian, a low-resource language with limited available training data. Due to this limitation, it is common practice to first pretrain multilingual models on high-resource languages such as English, and then fine-tune them on smaller languages. While the exact mechanisms behind this transfer are complex, the shared linguistic patterns among languages can enable the model to generalize better. In the case of Slovenian, additional training data from closely related South Slavic languages (such as Croatian, Bosnian, and Serbian) are often used to improve performance, as these languages share similar vocabulary [10].

Another important consideration is the choice between base models and instruction-tuned models. Base models, such as the original versions of GPT, T5, or BLOOM are trained to predict the next word in a sequence without any explicit understanding of task-specific instructions. As a result, they function more like autocomplete systems and are not suitable for tasks such as answering questions and summarizing. In contrast, instruction-tuned models, such as ChatGPT, FLAN-T5, or BLOOMZ have been further fine-tuned on datasets containing natural language instructions paired with expected outputs. This enables them to follow prompts like ”summarize this,” ”translate this,” or, in our case, ”generate a report from this data.” Since our task depends on instruction-driven behavior and will involve additional fine-tuning on report generation instructions, we focus our selection to mostly instruction-tuned models.

Table 1 lists the models selected as initial candidates for experimentation and potential fine-tuning, based on the outlined criteria.

Prompt Engineering

Before applying fine-tuning techniques, we will first explore whether Slovenian traffic report generation can be effectively solved using prompt engineering. We will employ role-based prompting, where we explicitly assign the model the role of a traffic reporter, and few-shot prompting, where we provide examples to guide the output. The structured prompt follows:

> *You are a traffic reporter. Generate a concise report based on the following data.*
>
> *Example 1: (Structured Data) → (Formatted Report)*
> *Example 2: (Structured Data) → (Formatted Report)*
> *Now generate a report for: (Structured Data)*

Additionally, we will explore step-by-step prompting, where the task is explicitly broken down into sequential steps, and chain-of-thought prompting, where the model is encouraged to reason through its output before generating a response.

Dataset

The dataset for fine-tuning our selected model is provided by RTV Slovenija and consists of structured traffic data paired with human-written reports. It includes a table with approximately 50 thousand entries, each describing real-time traffic incidents through 27 attributes such as affected roads, incident type, severity, and timestamps. Additionally, it contains around 28 thousand manually composed textual reports, which summarize traffic conditions in a concise and informative manner for radio broadcasting.

Implementation

Data Preprocessing

Our first step was to thoroughly examine the provided data and preprocess it to make it suitable for prompt engineering and potential fine-tuning. We found the raw data to be somewhat inconsistent and difficult to work with, which required significant cleaning and restructuring.

To standardize the input format, we removed the embedded markup language and created a unified .jsonl file. In this format, each entry groups all traffic events occurring between two reports. For example, events from 00:01 to 00:30 are assigned to the 00:30 report, while events after 00:30 are linked to the next one (e.g., 01:00).

In addition, we restructured the information by category, rather than relying solely on the A1 and B1 columns. This approach enables more structured inputs and could later allow us to identify which events were cancelled or are no longer valid. In doing so, we have laid the groundwork for generating even more structured representations—such as JSON objects or knowledge graphs that capture the overall state of Slovenian traffic.

Observing Base Models

To determine the most suitable model for further steps, including fine-tuning, we performed prompt engineering on four models: GaMS-1B, GaMS-9B, Llama 3.1 8B, and mt0-large. We began our exploration with a series of basic prompts to evaluate the models' general performance in simple, unrestricted scenarios. For instance, we used questions such as ”Kakšno je vreme danes?” and ”Kakšno je stanje v prometu?” to gauge their ability to provide relevant and coherent answers. While all models produced responses related to the questions, the information was fabricated, indicating a need to introduce structured input data.

In response, we provided the models with specific data to guide their answers. For example, we supplied weather information such as ”Povprečna temperatura v Sloveniji: 20 stopinj. Ljubljana: sončno, 22 stopinj.” and traffic data such as ”Na primorski avtocesti proti Ljubljani prometna nesreča. Nastaja zastoj.” This adjustment highlighted substantial differences in the models' behaviors and their ability to process and apply the given data.

GaMS-1B provided responses that were lengthy and verbose, often including additional irrelevant details unrelated to the input data, but are related to the topic. GaMS-9B displayed similar tendencies, generating extensive but largely irrelevant responses and occasionally introducing fictional information. Llama 3.1 8B demonstrated a partial ability to utilize the provided data but frequently produced overly generic or nonsensical outputs, revealing a lack of precision. Also Llama model frequently generated sentences that were grammatically incorrect in Slovenian. mt0-large, while concise, struggled with coherence and repeated phrases, as seen in its output for traffic data prompts. It sometimes struggled with Slovenian language.

We incorporated weather-related questions into our tests to observe how the models performed in other domains. This helped us evaluate their versatility and robustness in handling different types of queries.

Building on this initial evaluation, we proceeded with refining the prompt design and testing more sophisticated prompts. These new prompts included additional details and constraints specifically tailored for report generation. This iterative process allowed us to systematically enhance the models' ability to generate structured, accurate, and contextually appropriate outputs.

Prompt Engineering

After extensive testing and iterative refinements, we identified seven distinct prompts that provided the most promising results for generating structured, concise, and accurate traffic reports in Slovenian to use them in evaluation. Each prompt adhered to specific formatting and content requirements. Prompt 1 introduced a basic structure by providing an example groundtruth report along with the necessary data, focusing on producing concise summaries without redundancy. Prompt 2 emphasized a strict order of importance (wrong-way driving, accidents, traffic jams, road obstacles, and border crossings), with precise terminology for roads and directions, a six-sentence limit, specific sentence formulations, and prohibited words. Prompt 3 provided a model structure and examples of correct reports, ensuring events were sequenced by priority and avoiding repetition. Prompt 4 reinforced similar rules while adding detailed guidelines on road terminologies and standard expressions, requiring adherence to official naming conventions for Slovenian roads. This prompt offered the most comprehensive instructions and examples. Prompt 5 prioritized exact sequencing and formal descriptions of incidents using predefined sentence structures without providing examples. Prompt 6 focused on specific language rules, clarifying terms like ”slow lanes” and ”covered tunnels” to ensure professional phrasing and accurate geographical references, supported by two examples of data and correct reports. Lastly, Prompt 7, written in English but requiring Slovenian reports, mandated the use of only two predefined sentence structures, eliminating non-essential phrasing and repetitive details while maintaining clarity.

The evaluation process involved testing these seven prompts with three data formulations. First, the complete dataset for a single report was used, obtained from the preprocessing step. Second, reports were generated based on one piece of information at a time. Finally, clean data without repeated information or additional tags about the event (e.g., ”OVIRE,” ”ZASTOJI,” ”MEJNI PREHODI,” ”POMEMBNO”) was tested. This comprehensive approach ensured a thorough assessment of each prompt’s effectiveness.

Evaluation

Empirical evaluation

To better understand how language models handle the generation of traffic reports under varying data quality and formatting constraints, we evaluated the four models (GaMS-1B, GaMS-9B, Llama 3.1 8B, and mt0-large) on seven distinct above-described prompts.

The models showed clear differences in performance when generating traffic reports. Llama 3.1 8B performed best overall, producing well-structured and informative outputs with the fewest grammatical errors, though it occasionally introduced unnecessary information or repeated content. GaMS-1B and GaMS-9B both struggled with relevance and information fidelity. GaMS-1B often produced overly brief reports missing key details, while GaMS-9B tended toward verbose and off-topic outputs, sometimes inventing events and showing coherence issues. mt0-large performed poorly, frequently generating irrelevant, repetitive, or nonsensical content regardless of prompt design. Only Llama responded reliably to prompts with structured formatting and demonstrated some consistency in following language and structure requirements. Across all models, prompt specificity was crucial - especially for GaMS models, which required very detailed instructions to achieve partial compliance.

The evaluation across various prompt types revealed consistent performance patterns among the models. Llama 3.1 8B performed best in structured prompts with strict formatting requirements (Prompts 2, 4, 6, 7), generally adjusting to the defined hierarchies and traffic terminology. However, it occasionally missed important elements or relevant details when overly constrained by formatting requirements. In contrast, the GaMS models struggled across the board — particularly in open-ended prompts (Prompts 1, 3, 5), producing verbose, often irrelevant outputs, frequently ignoring instructions and making events up, or including unrelated information. On the other hand, GaMS-9B produced more fluid and natural-sounding text, which could be beneficial in open-ended prompts. mt0-large consistently fails to meet even basic task requirements, either by copying prompt instructions or generating incoherent and unusable text. The findings emphasize that structured prompting improves output reliability, but also highlight the need for targeted fine-tuning to address omissions, made-up information, and grammatical correctness. These limitations are especially critical for real-world deployment, where clarity and precision are essential.

Evaluation using metrics

We employed the BLEU (Bilingual Evaluation Understudy) metric to quantitatively evaluate the quality of generated traffic reports. BLEU measures the similarity between model-generated text and human reference translations by comparing their n-gram patterns. Our implementation assessed matches from 1-grams (individual words) to 4-grams (four-word phrases), with equal weighting for each n-gram level. The metric aligns with traffic reporting needs, where both keyword accuracy (e.g., location names) and multi-word expressions (e.g., ”zaprt vozni pas”) are critical.

Table 2. BLEU scores for different models using Prompt 7.

Model	BLEU
GaMS-1B	0.050
GaMS-9B	0.061
Llama-3.1-8B	0.205
mt0-large	0.041

The BLEU scores (Table 2) support our empirical findings, with Llama 3.1 8B clearly outperforming the other models. Its score of 0.205 reflects stronger compliance with prompt instructions and better structure compared to other models. GaMS-9B slightly outperforms GaMS-1B, though both remain low overall. Notably, GaMS-9B reached its peak performance of 0.11 on Prompt 2, which involved more structured guidance. This suggests that GaMS-9B may show some potential under stricter prompting or with targeted fine-tuning. mt0-large scores the lowest, reinforcing its observed failure to generate coherent or relevant content. These results suggest Llama 3.1 8B or GaMS-9B hold the most promise for generating traffic reports but requires fine-tuning, while the other models show limited suitability for this task in their current forms.

Future Work

Challenges

A significant challenge with generating traffic reports using LLMs stems from inconsistencies in the provided data from RTV Slovenia. In the Excel data sheets, some critical information is occasionally missing. For example, there are cases where the data indicates waiting times at border crossings (”na mejnem prehodu je čakalna doba”) without specifying the duration of the wait. In other instances, entire events are missing from the dataset altogether. Conversely, some events recorded in the Excel table for a given time window are not mentioned in the corresponding traffic report.

Another issue arises from reports including events that occurred in previous time frames but should now be canceled in the current report. To address these challenges, further data preprocessing steps will be necessary to align events with their appropriate time frames and ensure that all relevant information is accurately reflected in reports. This includes handling incomplete data, reconciling discrepancies between the dataset and the reports, and ensuring that events from previous time frames are appropriately updated or removed.

Structured Data Integration

To address data inconsistencies, we propose integrating structured inputs such as JSON schemas or knowledge graphs to represent traffic events in a consistent format. These structures allow for clear definition of event attributes (e.g., location, duration, type) and relationships, making it easier to manage event states—such as identifying and marking cancelled or outdated entries.

Fine-Tuning

Our results show that while prompt engineering played a significant role in shaping the performance of the models, next steps like fine-tuning are crucial to improve the results. Llama 3.1 8B and GaMS-9B, while the strongest performers overall, still exhibited omissions and inconsistencies in details that could be addressed through fine-tuning on high-quality and structured traffic data. Another idea could involve fine-tuning the model with a mixture of structured and semi-structured prompts to better simulate real-world scenarios where reporting needs can vary in complexity. Prompts that offer some flexibility in structure could generate more complete and contextually accurate traffic reports. In conclusion, focusing on fine-tuning Llama 3.1 8B or GaMS-9B while refining prompt engineering seems to be the best approach to enhance its performance.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[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] Jinhui Ouyang, Yijie Zhu, Xiang Yuan, and Di Wu. Trafficgpt: Towards multi-scale traffic analysis and generation with spatial-temporal agent framework, 2024.

[5] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with entity modeling, 2019.

[6] Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures, 2019.

[7] Kancharla Aditya Hari, Manish Gupta, and Vasudeva Varma. Curriculum learning for cross-lingual data-to-text generation with noisy data, 2024.

[8] Xuanyi Zhang, Genghong Zhao, Yi Ren, Weiguang Wang, Wei Cai, Yan Zhao, Xia Zhang, and Jiren Liu. Data augmented large language models for medical record generation. *Applied Intelligence*, 55(2):1–22, 2025.

[9] Domen Vreš, Martin Božič, Aljaž Potočnik, Tomaž Martinčič, Iztok Lebar Bajec, Timotej Petrič, and Marko Robnik-Šikonja. GaMS-1B-Chat: Generative model for slovene. https://huggingface.co/cjvt/GaMS-1B-Chat, 2024. University of Ljubljana and XLAB. Instruction-tuned version of GaMS developed within the PoVeJMo research program.

[10] Domen Vreš, Martin Božič, Aljaž Potočnik, Tomaž Martinčič, and Marko Robnik-Šikonja. Generative model for less-resourced language with 1 billion parameters. In *Proceedings of the Conference on Language Technologies and Digital Humanities*, Ljubljana, Slovenia, 2024. Znanstvena založba Filozofske fakultete, Univerza v Ljubljani.