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

Iterative Refinement of LLM Instructions for Automated Slovenian Traffic News Generation for RTV Slovenija

NLP Project Group

Abstract

This project addresses the automatic generation of Slovenian traffic news, specifically targeting the needs of RTV Slovenija, where such reports are currently manually prepared. Utilizing Excel data from the `promet.si` portal, the system employs a two-Large Language Model (LLM) approach. The Slovenian GaMS-9B-Instruct model generates initial traffic reports based on engineered prompts and specific guidelines. Subsequently, the Gemini 2.5 flash-preview model evaluates these reports and iteratively refines the instructions provided to GaMS. This methodology focuses on enhancing the quality, accuracy, and adherence of LLM-generated news to RTV Slovenija's stylistic and content requirements through a feedback loop, offering an alternative to direct LLM fine-tuning for this task. The core contribution is an adaptable pipeline demonstrating LLM-driven instruction improvement for specialized Slovenian text generation.

Keywords

Natural Language Processing, Large Language Models, Instruction Tuning, Prompt Engineering, Traffic Information Systems, Text Generation, Automated Journalism, Slovenian Language, RTV Slovenija

Advisors: Prof. Slavko Žitnik

Introduction

The timely and accurate dissemination of traffic information is crucial for road users, with RTV Slovenija radio presenters currently relying on manually checked and typed reports. This project aims to automate the generation of these short, regular, and important Slovenian traffic news reports. The primary data source is an Excel file containing information from the `promet.si` portal.

The core challenge lies in ensuring the generated news adheres to specific guidelines and instructions regarding content, style, and priority, as required by RTV Slovenija. While LLM fine-tuning is one approach, this project explores an alternative strategy: leveraging advanced prompt engineering techniques with an existing Slovenian LLM (GaMS-9B-Instruct [1]) and employing a second, more powerful LLM (Gemini 2.5 flash-preview [2]) for iterative evaluation and refinement of the instructions given to the primary generation model.

The relevance of this work is in developing a system that can adapt and improve its output quality over time through an LLM-driven feedback mechanism, potentially reducing the need for extensive manual oversight or large datasets typically required for fine-tuning. The project's goal is to create a pipeline that:

1. Extracts and preprocesses traffic data from the provided Excel source.
2. Generates coherent Slovenian traffic news reports using GaMS-9B-Instruct, guided by an evolving set of instructions.
3. Utilizes Gemini 2.5 flash-preview to assess GaMS's adherence to these instructions and to propose targeted revisions, thereby enhancing future report quality.

This iterative refinement process is central to tailoring the system's output to meet the detailed specifications of RTV Slovenija's traffic reporting.

Methods

The system's methodology aligns with several aspects of the original project guidelines, focusing on prompt engineering and a novel evaluation-refinement loop.

Literature Review and Data Exploration (Methodology 6)

A review of available LLMs led to the selection of cjvt/GaMS-9B-Instruct for Slovenian text generation due to its specific training on Slovenian data, and Gemini 2.5 flash-preview for

its advanced reasoning and instruction-following capabilities, making it suitable for evaluation and meta-instruction tasks. The provided Excel data (`./Data/RTVSlo/Podatki - PrometnoPorocilo_2022_2023_2024.xlsx`) was analyzed, leading to the development of the get_final_traffic_text function in Data/readData.py. This function performs:

- Reading data from the relevant year's sheet using Pandas [3].
- Filtering entries within a 15-minute window of a user-specified timestamp.
- Consolidating text from columns 'A1', 'B1', 'C1' (and 'A2', 'B2', 'C2').
- Using Levenshtein distance [4] (threshold ≥ 0.85) to select the most representative report if multiple similar ones exist, or the latest otherwise.
- Cleaning HTML content with BeautifulSoup [5] and basic text normalization.

Initial Solution: Prompt Engineering (Methodology 7)

The initial approach to solving the task relied exclusively on prompt engineering for the GaMS-9B-Instruct model. A comprehensive set of `default_custom_instructions` was developed (as seen in `main.py`). These instructions covered:

- Urgency-based ordering of news items.
- Specific sentence structures.
- Consistent naming conventions for motorways and directions.
- Rules for excluding minor traffic jams.
- Special handling for road closures (obvozi), event conclusions (opoved), and weather conditions like burja.

This formed the baseline for GaMS's generation capabilities.

Evaluation Definition and Implementation (Methodology 8)

A semi-automatic evaluation mechanism was implemented using the Gemini 2.5 flash-preview model. The chat_with_gemini function in LLMs/gemy.py defines this process:

- Gemini receives: the instructions given to GaMS, the raw input traffic data, and GaMS's generated news report.
- It is prompted with a meta-instruction to:
    1. Assess GaMS's adherence to each specific instruction point. This implicitly covers criteria like "identification of important news" (via urgency rules), "correct roads namings," "correct filtering" (via inclusion/exclusion rules), and appropriateness of "text lengths and words."
    2. List instructions followed and not followed.
    3. Provide a numerical adherence score (0-1).
    4. Propose revised instructions for GaMS, prefixed with a '$' character for extraction.

This evaluation is "semi-automatic" as the user ultimately decides whether to adopt Gemini's suggested instruction changes. Gemini's score serves as a direct performance metric for instruction adherence.

LLM Interaction for Improvement (Alternative to Fine-tuning - Methodology 9)

Instead of direct parameter-efficient fine-tuning of GaMS, this project focused on improving its performance through an iterative instruction refinement loop orchestrated by `main.py` and involving two LLMs:

1. Traffic News Generation (GaMS-9B-Instruct): Managed by LLMs/gaMS.py, this module initializes a Hugging Face transformers pipeline for cjvt/GaMS-9B-Instruct. It supports GPU usage with optional 4-bit quantization [6, 7]. The chat_with_gams function generates news based on the current (potentially refined) instructions and input data, maintaining a conversation history. Key parameters: max_new_tokens=2048, do_sample=True, temperature=0.7.

2. Instruction Refinement and Evaluation (Gemini 2.5 flash-preview): As described above, LLMs/gemy.py uses Gemini to evaluate GaMS's output and suggest better instructions. Generation parameters for Gemini: max_output_tokens=9000, temperature=0.1.

3. Iterative Loop: The `main.py` script facilitates user interaction:

1. User provides a date-time.
2. Processed traffic data is fed to GaMS with current instructions.
3. GaMS's output is evaluated by Gemini, which suggests new instructions.
4. The user decides if these new instructions should replace the old ones for subsequent iterations.

This provides an interactive test interface and a mechanism for progressive improvement.

Performance Analysis (Methodology 10)

The effectiveness of the iterative refinement technique is assessed by:

- Tracking changes in Gemini's adherence score for GaMS's output across iterations.
- Qualitatively analyzing the differences in GaMS-generated text, focusing on improvements in areas like urgency ordering, stylistic consistency, and adherence to specific formatting rules.
- Implicit human evaluation occurs when the user decides whether Gemini's suggested instruction changes are beneficial and should be adopted.

While standard metrics like Precision, Recall, and F1 were not directly implemented for output text against a gold standard, Gemini's structured feedback and score provide a measure of how well the generated news aligns with the desired characteristics defined in the instructions.

Results

The iterative refinement process was tested using sample traffic data (timestamp: 2024-01-02 08:17:07), which included events like a broken-down vehicle, HGV restrictions, fog, mandatory snow chains, and a planned road closure. The goal was to observe how instruction changes suggested by Gemini improved GaMS's adherence to RTV Slovenija's reporting guidelines.

Initial Output and Evaluation (Iteration 0)

Using the `default_custom_instructions`, GaMS produced a narrative report in the past tense (e.g., *"This morning on the radio we reported..."*).

Gemini's evaluation:

- **Complied with (examples):** Sentence structure, basic road naming, detour information.
- **Not complied with (examples):** Urgency order (Rule 1 - started with broken vehicle (priority 7) instead of the planned major road closure (priority 5 if considered as such, or handled as a general announcement)), live radio style (used past tense).
- **Score: 0.55**

Gemini proposed new instructions emphasizing a "live" reporting style and stricter adherence to urgency.

Refined Output and Evaluation (Iteration 1)

After the user accepted Gemini's refined instructions, GaMS processed the same input: The output shifted to a more direct, present-tense style, but introduced unrequested category headings (e.g., *"**Primorska highway** - A broken-down vehicle is obstructing traffic..."*) and still did not perfectly follow the complex urgency ordering.

Gemini's evaluation:

- **Complied with (examples):** Improved sentence structure for live feel, road naming.
- **Not complied with (examples):** Strict urgency order still not perfectly met, introduction of unsolicited category headers, inclusion of a non-news closing remark (from input data).
- **Score: 0.65**

The score improved, indicating better alignment with some refined points (e.g., tone). Gemini's next set of suggested instructions would likely target the new issues like forbidding headers and re-emphasizing the event ordering and filtering of non-essential concluding remarks.

Summary of Iterative Improvement

The process demonstrates how targeted feedback can guide the generation model. Table 1 summarizes this.

The improvement in score and qualitative aspects (like reporting style) shows the potential of this iterative method for enhancing adherence to complex guidelines like correct filtering of information, appropriate text length/wording (by following structural rules), and identification of important news (through urgency rules).

Table 1. Iterative Improvement of GaMS Output through Gemini Feedback, focusing on adherence to RTV Slovenija guidelines.

Iteration	Key Instruction Change	GaMS Output	Score
0 (Initial)	Baseline instructions	Past tense, incorrect urgency	0.55
1	Emphasize "live" style, strict urgency	More direct tone, but still order/formatting issues	**0.65**

Discussion

This project successfully implemented an iterative system for refining LLM instructions to automate the generation of Slovenian traffic news for RTV Slovenija. The chosen two-LLM approach, with GaMS-9B-Instruct for generation and Gemini 2.5 flash-preview for evaluation and instruction refinement, offers a practical alternative to direct model fine-tuning for enhancing adherence to specific domain requirements.

Effectiveness of the Approach

The system demonstrated its ability to progressively improve the quality of generated news reports. Gemini's feedback loop allowed for targeted adjustments to GaMS's instructions, leading to observable improvements in output style, tone, and partial adherence to content rules. This method of "teaching" the generation LLM through evolving prompts, guided by another LLM, is a key strength, especially when large, high-quality datasets for fine-tuning are unavailable or when rapid adaptation to changing guidelines is needed. The interactive nature allows a human-in-the-loop to validate the refinement process.

Relation to Original Project Goals

While the original project description mentioned "fine-tuning" an LLM, this work focused on sophisticated prompt engineering and an LLM-driven instruction refinement cycle as the primary means of improvement. This aligns with "leveraging prompt engineering techniques" and defining "(semi-)automatic evaluation criteria." The system provides an "interface to do an interactive test" for the generation and refinement process. The evaluation, while not using P/R/F1 against a gold corpus, uses Gemini's scoring and qualitative feedback to assess performance regarding "identification of important news, correct roads namings, correct filtering, text lengths and words."

Limitations and Challenges

- **Evaluator LLM Dependency:** The success hinges on Gemini's capability to accurately interpret GaMS's output against complex instructions and to generate effective, non-conflicting new instructions.
- **Generator LLM Adherence:** GaMS may not always perfectly follow even well-crafted instructions, particularly highly nuanced or multi-conditional rules like strict urgency ordering across diverse event types.
- **Instruction Complexity:** Iteratively refined instructions can become very long and potentially introduce subtle contradictions if not carefully managed by the evaluating LLM or a human overseer.
- **Scope of Evaluation:** Gemini's evaluation is based on the provided instructions. It doesn't inherently measure factual accuracy against the real world, only consistency with the input data as interpreted through the instructions.

Future Work

- **Hybrid Approach:** Use the outputs generated from several successful refinement iterations as a high-quality dataset to perform parameter-efficient fine-tuning on GaMS. This could help internalize the desired behaviors more robustly.
- **Automated Instruction Acceptance Logic:** Develop criteria (e.g., consistent score improvement over N iterations, specific error type reduction) for automatically accepting Gemini's instruction changes to reduce manual intervention.
- **More Granular Evaluation:** Enhance Gemini's evaluation prompt to provide more structured feedback, perhaps scoring adherence to individual rules or categories of rules.
- **Comparative Study:** Formally compare this iterative instruction refinement method with direct few-shot prompting (with many examples) or a baseline fine-tuning approach on a small, curated dataset.
- **User Interface for RTV Slovenija:** Develop a more user-friendly interface for RTV Slovenija personnel to use the system, review generated news, and perhaps provide direct feedback into the refinement loop.

Conclusion

The project demonstrates a viable and adaptive method for generating specialized Slovenian traffic news reports by iteratively refining instructions for a generation LLM using feedback from an evaluation LLM. This approach effectively addresses many of the complexities involved in tailoring LLM outputs for specific, rule-intensive domains like traffic reporting for RTV Slovenija, offering a promising direction for developing more controllable and accurate automated content generation systems.

References

[1] CJVT. Gams-9b-instruct model card. https://huggingface.co/cjvt/GaMS-9B-Instruct, 2023. Accessed May 2025.

[2] Google. Gemini api documentation. https://ai.google.dev/docs/gemini_api_overview, 2024. Accessed May 2025.

[3] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[4] Sten Helmuth. python-levenshtein. https://github.com/ztane/python-Levenshtein/, 2023. Accessed May 2025.

[5] Leonard Richardson. Beautiful soup documentation. https://www.crummy.com/software/BeautifulSoup/bs4/doc/, 2024. Accessed May 2025.

[6] Tim Dettmers et al. bitsandbytes. https://github.com/TimDettmers/bitsandbytes, 2023. Accessed May 2025.

[7] Hugging Face. Accelerate: A simple way to train and use pytorch models with multi-gpu, tpu, mixed-precision. https://github.com/huggingface/accelerate, 2024. Accessed May 2025.