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

Anže Hočevar in Jan Anžur

Abstract

The abstract goes here. The abstract goes here.

Keywords
Keyword1, Keyword2, Keyword3 ...

Advisors: Slavko Žitnik

Introduction

Automated news generation (often called automated journalism or robot journalism) refers to software producing news from data with minimal human input [1]. Early NLG systems in newsrooms were largely template-based, using predefined phrases filled with data (e.g., finance or sports reports) [1]. Recent advances in deep learning and especially the Transformer architecture have greatly improved the ability to generate coherent, contextually accurate narratives from structured data [2]. For example, Leppänen et al. (2017) developed a data-driven NLG system that produced thousands of localized election news articles in multiple languages [1]. These systems highlight the potential of automated content creation, but also underscore requirements like transparency, accuracy, and adaptability in journalistic contexts [1]. Ensuring factual correctness and neutrality is paramount – research notes that maintaining objectivity and avoiding bias remain significant challenges for AI-generated news [3]. Modern LLMs, with their ability to generate fluent text, are now being explored as core engines for such NLG tasks.

Literature Review

One of the key considerations in automating Slovenian traffic news is the availability and adaptability of LLMs for multilingual text generation. Studies have shown that most state-of-the-art models, including OpenAI's GPT series [4], BLOOM [5], and mT5 [6], demonstrate strong multilingual capabilities. However, Slovenian, being a low-resource language, remains under-represented in large-scale training corpora [7]. Locally fine-tuned models such as SloT5 [8] have emerged to address this gap, showing promise in domain-specific Slovenian text generation.

A crucial challenge in this domain is balancing between **fine-tuning** and **prompt engineering**. Fine-tuning LLMs on domain-specific text can improve accuracy but requires computational resources and well-annotated data [9]. Prompt engineering, on the other hand, provides a lighter-weight alternative by designing effective input prompts to guide the model's response [10]. For structured and time-sensitive content like traffic news, a hybrid approach may be necessary, combining a base model with well-optimized prompting techniques.

Additionally, context-aware traffic reporting can benefit from external data sources, such as live weather updates, public holiday schedules, and road congestion analytics. Studies have indicated that integrating real-time sensor data and probabilistic event modelling improves the predictive accuracy of automated reports [11].

Based on these insights, the following sections analyse the provided structured dataset of Slovenian traffic news, examining patterns in report generation and urgency to inform an AI-based automation approach.

Methods

Data

We are working with traffic news data. More specifically, we have a set of traffic news reports that serve as a part of our training data. The other part is hidden in the periodically scraped website data that in some way or another corresponds to parts of the traffic news reports. The input to our NLP model will be unseen scraped website data, whereas the output will be a report that follows the intended format.

Models

We used an open-source large language model (LLM) named GaMS-27B-Instruct[1] to generate new data. Since our data is basically completely in Slovenian, we needed an LLM that was trained specifically in this language. Technically speaking, it is a fine-tuned version of the Google Gemma2 LLM[2], so it performs well both on English and Slovenian.

Preprocessing pipeline

The raw input consists of structured traffic event logs in Excel format provided by the `promet.si` portal, and human-written traffic reports in RTF format, broadcast by RTV Slovenija. To align these for training and evaluation, we implemented the following preprocessing pipeline:

1. Time-based Filtering. To align structured input data with its corresponding RTF report, we apply a temporal filtering step. Specifically, we extract all rows from the Excel dataset whose timestamps fall within a fixed window *prior* to the timestamp of the selected RTF report (typically 1 to 8 hours). This windowed extraction is necessary because RTV traffic reports do not reflect only newly observed events, but also often reiterate unresolved or ongoing events from previous intervals. For example, a road closure or prolonged congestion may appear in multiple consecutive reports until the situation is resolved. By including several hours of prior data, we ensure that the input context is sufficient to reconstruct both new and persisting traffic conditions reflected in the human-authored report.

2. Cleaning and Normalization. All text fields undergo normalization using HTML stripping (via BeautifulSoup), whitespace collapsing, and newline removal. Timestamp fields are converted to `datetime` objects for temporal grouping.

3. Sentence Extraction and Deduplication. Each row is processed into individual sentences. Sentences are cleaned and deduplicated using one of several strategies:

- **Exact deduplication:** Removes literal duplicates across rows.
- **Semantic deduplication:** Uses Sentence-BERT embeddings and cosine similarity to group semantically similar sentences (> 0.7) and retain a representative.

4. Content Selection Strategies. We evaluate multiple strategies to retain the most relevant information:

- **Longest informative sentence:** Keeps the longest sentence within a semantic group.
- **TF-IDF selection:** Scores sentences using TF-IDF and selects the highest-scoring one per group.
- **Named entity preference:** Prefers sentences containing domain-specific keywords (e.g., road names, locations, traffic event types).

5. Flat Input Construction. Once the relevant and deduplicated sentences are selected, they are assembled into a single paragraph to be used as the prompt input for language model generation. We explored several formatting strategies to mimic the structure and tone of actual RTV Slovenija traffic broadcasts:

a) **Basic concatenation.** All selected sentences are simply joined with a space delimiter to form one long paragraph that still includes html tags. This approach ensures full coverage of the input without introducing artificial breaks or sectioning.

b) **Structured template with RTV header.** The paragraph is prefixed with a timestamp header consistent with RTV Slovenija formatting:

```
Prometne informacije    DD. MM.
YYYY    HH.MM    2. program
```

followed by a standard opening:

```
Podatki o prometu.
```

This approach makes the output stylistically and structurally comparable to the ground truth RTF reports, enabling more robust evaluation.

The structured variant (b) most closely resembles the output format expected in downstream evaluation and human interpretation. It might be also more suitable for fine-tuning models in tasks such as summarization or report generation.

6. RTF Report Matching. The RTF file closest in time to the target timestamp is parsed and used as the reference summary. Date and time are extracted from the header line using regular expressions.

This preprocessing pipeline enables consistent comparison between structured data and human-written summaries, and prepares high-quality input for evaluation or language model fine-tuning.

[1] https://huggingface.co/cjvt/GaMS-27B-Instruct

[2] https://huggingface.co/google/gemma-2-27b

Results

Baseline

As a starting point, we attempted to approach this task by simply using just common prompt engineering methods. To put it more bluntly, we fed the LLM with a prompt that concisely and clearly describes the task at hand and provided a sequence of shots (examples) that it should be able to follow. An example of this approach in actions is shown in figure 1. It is apparent that the output is very similar to the input. Although, this is not necessarily bad, as long as the rules for data generation are not violated.

Discussion

Use the Discussion section to objectively evaluate your work, do not just put praise on everything you did, be critical and exposes flaws and weaknesses of your solution. You can also explain what you would do differently if you would be able to start again and what upgrades could be done on the project in the future.

Acknowledgments

Here you can thank other persons (advisors, colleagues ...) that contributed to the successful completion of your project.

References

[1] ACL Anthology. Advances in automated journalism and natural language generation. *ACL Conference*, 2023.

[2] Arxiv.org. Transformer-based deep learning for natural language processing. *arXiv preprint*, 2022.

[3] Arxiv.org. Bias and objectivity challenges in ai-generated news. *arXiv preprint*, 2023.

[4] Tom Brown and et al. Language models are few-shot learners. *NeurIPS*, 2020.

[5] Teven Scao and et al. Bloom: A 176b parameter open-access multilingual language model. *arXiv preprint*, 2022.

[6] Linting Xue and et al. mt5: A massively multilingual pre-trained text-to-text transformer. *Journal of Machine Learning Research*, 2021.

[7] Matej Ulčar and Marko Robnik-Šikonja. Sloberta: The slovenian roberta model. *arXiv preprint*, 2021.

[8] Matej Ulčar and Marko Robnik-Šikonja. Slot5: Pre-trained text-to-text models for slovenian. *arXiv preprint*, 2022.

[9] Y. Zhang and et al. Fine-tuning strategies for large language models. *Journal of Artificial Intelligence Research*, 2022.

[10] L. Reynolds and K. McDonell. Prompt programming for large language models. *arXiv preprint*, 2021.

[11] Sebastian Schelter and et al. Automated event-driven news generation. *Data Mining and Knowledge Discovery*, 2019.

```
VHOD:
<p><strong>Zastoji</strong></p>
<p>- na ljubljanski južni obvoznici od Rudnika proti razcepu Kozarje;
    <br>- na gorenjski avtocesti od Kosez do predora Šentvid proti Kranju;
    <br>- na štajerski avtocesti med Blagovico in Trojanami proti Celju ter pred
     ↪  prehodom Šentilj proti Avstriji;
    <br>- na ljubljanski zahodni obvoznici od Kosez proti Kozarjam ter na primorski
     ↪  avtocesti med Ljubljano in Brezovico proti Kopru;
    <br>- na cestah Ljubljana - Brezovica, Lesce - Bled in Šmarje - Koper.
</p>

IZHOD:
Zastoji so na cestah Šmarje-Koper in Lesce-Bled.
Prav tako so zastoji na zahodni in južni ljubljanski obvoznici od Kosez in Rudnika
 ↪  proti Primorski. Potovalni čas se na obeh odsekih podaljša za približno 15 minut.
Zaradi del je krajši zastoj na štajerski avtocesti med Blagovico in Trojanami proti
 ↪  Mariboru. Opozarjamo na nevarnost naleta.
```

Figure 1. An example of LLM output generation for given input data using only prompt engineering methods.