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 large lanugage models (LLMs), with their ability to generate fluent text, are now being explored as core engines for such NLG tasks.

This dataset in question for this paper are short traffic news reports. It includes periodically sampled data from the website of the Slovenian national motorway management company and reports that are broadcasted on a public radio station. The objective is to enhance an LLM's ability to generate reports that align seamlessly with the style and structure of the provided data by utilizing the appropriate natural language processing techniques.

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.

Parthasarathy et al. propose and describe a seven stage pipeline for fine-tuning an LLM that covers everything from data preparation to model evaluation [11]. It also provides an overview of (parameter-efficient) fine-tuning techniques, en-

abling the selection of the optimal approach based on specific performance objectives and time efficiency requirements.

Chen and Xiao explore a unique news summary generator that works by utilizing a fine-tuned LLM to generate a pool of structured event patterns, then applying a genetic algorithm on them to heuristically find the best ones and finally using another LLM to generate the summaries from those event patterns [12].

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 [13]. One option to explore is to include such data in the prompt at runtime, resembling a retrieval-augmented generation (RAG) chain approach, only with live data instead of extensive documents.

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

Use the Methods section to describe what you did an how you did it – in what way did you prepare the data, what algorithms did you use, how did you test various solutions ... Provide all the required details for a reproduction of your work.

Below are LaTeXexamples of some common elements that you will probably need when writing your report (e.g. figures, equations, lists, code examples ...).

Equations

You can write equations inline, e.g. $\cos\pi = -1$, $E = m \cdot c^2$ and α, or you can include them as separate objects. The Bayes's rule is stated mathematically as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{1}$$

where A and B are some events. You can also reference it – the equation 1 describes the Bayes's rule.

Lists

We can insert numbered and bullet lists:

1. First item in the list.
2. Second item in the list.
3. Third item in the list.

- First item in the list.
- Second item in the list.
- Third item in the list.

We can use the description environment to define or describe key terms and phrases.

Word What is a word?.

Concept What is a concept?

Idea What is an idea?

Random text

This text is inserted only to make this template look more like a proper report. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam blandit dictum facilisis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam convallis tellus velit, quis ornare ipsum aliquam id. Maecenas tempus mauris sit amet libero elementum eleifend. Nulla nunc orci, consectetur non consequat ac, consequat non nisl. Aenean vitae dui nec ex fringilla malesuada. Proin elit libero, faucibus eget neque quis, condimentum laoreet urna. Etiam at nunc quis felis pulvinar dignissim. Phasellus turpis turpis, vestibulum eget imperdiet in, molestie eget neque. Curabitur quis ante sed nunc varius dictum non quis nisl. Donec nec lobortis velit. Ut cursus, libero efficitur dictum imperdiet, odio mi fermentum dui, id vulputate metus velit sit amet risus. Nulla vel volutpat elit. Mauris ex erat, pulvinar ac accumsan sit amet, ultrices sit amet turpis.

Phasellus in ligula nunc. Vivamus sem lorem, malesuada sed pretium quis, varius convallis lectus. Quisque in risus nec lectus lobortis gravida non a sem. Quisque et vestibulum sem, vel mollis dolor. Nullam ante ex, scelerisque ac efficitur vel, rhoncus quis lectus. Pellentesque scelerisque efficitur purus in faucibus. Maecenas vestibulum vulputate nisl sed vestibulum. Nullam varius turpis in hendrerit posuere.

Figures

You can insert figures that span over the whole page, or over just a single column. The first one, Figure 1, is an example of a figure that spans only across one of the two columns in the report.

On the other hand, Figure 2 is an example of a figure that spans across the whole page (across both columns) of the report.

Tables

Use the table environment to insert tables.

Table 1. Table of grades.

Name		
First name	Last Name	Grade
John	Doe	7.5
Jane	Doe	10
Mike	Smith	8

Code examples

You can also insert short code examples. You can specify them manually, or insert a whole file with code. Please avoid

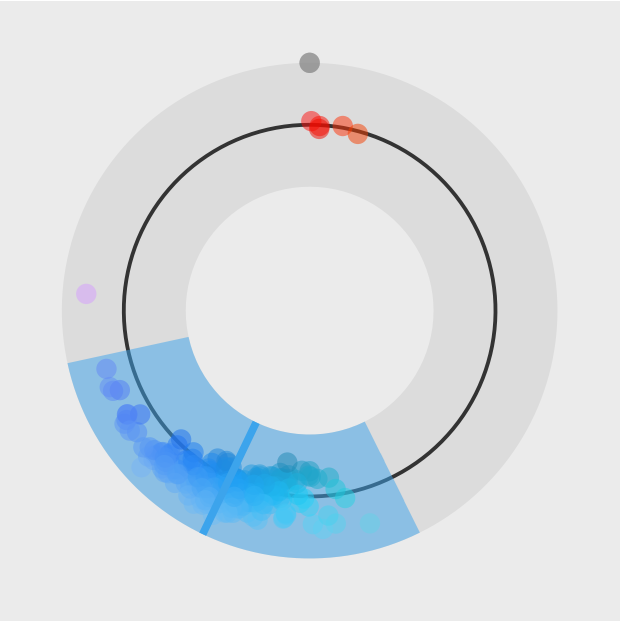

Figure 1. A random visualization. This is an example of a figure that spans only across one of the two columns.

inserting long code snippets, advisors will have access to your repositories and can take a look at your code there. If necessary, you can use this technique to insert code (or pseudo code) of short algorithms that are crucial for the understanding of the manuscript.

Listing 1. Insert code directly from a file.

```
import os
import time
import random

fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

Listing 2. Write the code you want to insert.

```
import(dplyr)
import(ggplot)

ggplot(diamonds,
           aes(x=carat, y=price, color=cut)) +
  geom_point() +
  geom_smooth()
```

Results

Use the results section to present the final results of your work. Present the results in a objective and scientific fashion. Use visualisations to convey your results in a clear and efficient manner. When comparing results between various techniques use appropriate statistical methodology.

More random text

This text is inserted only to make this template look more like a proper report. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam blandit dictum facilisis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam convallis tellus velit, quis ornare ipsum aliquam id. Maecenas tempus mauris sit amet libero elementum eleifend. Nulla nunc orci, consectetur non consequat ac, consequat non nisl. Aenean vitae dui nec ex fringilla malesuada. Proin elit libero, faucibus eget neque quis, condimentum laoreet urna. Etiam at nunc quis felis pulvinar dignissim. Phasellus turpis turpis, vestibulum eget imperdiet in, molestie eget neque. Curabitur quis ante sed nunc varius dictum non quis nisl. Donec nec lobortis velit. Ut cursus, libero efficitur dictum imperdiet, odio mi fermentum dui, id vulputate metus velit sit amet risus. Nulla vel volutpat elit. Mauris ex erat, pulvinar ac accumsan sit amet, ultrices sit amet turpis.

Phasellus in ligula nunc. Vivamus sem lorem, malesuada sed pretium quis, varius convallis lectus. Quisque in risus nec lectus lobortis gravida non a sem. Quisque et vestibulum sem, vel mollis dolor. Nullam ante ex, scelerisque ac efficitur vel, rhoncus quis lectus. Pellentesque scelerisque efficitur purus in faucibus. Maecenas vestibulum vulputate nisl sed vestibulum. Nullam varius turpis in hendrerit posuere.

Nulla rhoncus tortor eget ipsum commodo lacinia sit amet eu urna. Cras maximus leo mauris, ac congue eros sollicitudin ac. Integer vel erat varius, scelerisque orci eu, tristique purus. Proin id leo quis ante pharetra suscipit et non magna. Morbi in volutpat erat. Vivamus sit amet libero eu lacus pulvinar pharetra sed at felis. Vivamus non nibh a orci viverra rhoncus sit amet ullamcorper sem. Ut nec tempor dui. Aliquam convallis vitae nisi ac volutpat. Nam accumsan, erat eget faucibus commodo, ligula dui cursus nisi, at laoreet odio augue id eros. Curabitur quis tellus eget nunc ornare auctor.

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.

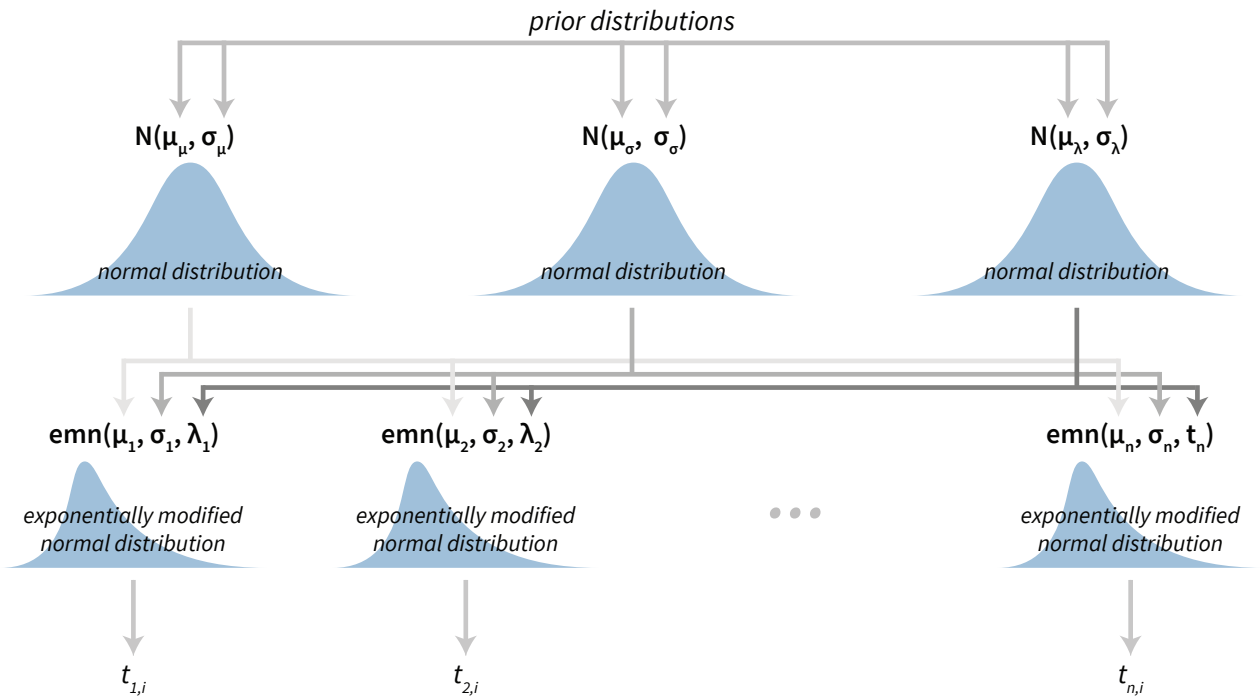

Figure 2. Visualization of a Bayesian hierarchical model. This is an example of a figure that spans the whole width of the report.

[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] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*, 2024.

[12] Le Xiao and Xiaolin Chen. Enhancing llm with evolutionary fine tuning for news summary generation. *arXiv preprint arXiv:2307.02839*, 2023.

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