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

Automatic generation of Slovenian traffic news for RTV Slovenija

Vid Kobal, Matija Tomažič, and Žan Luka Nusdorfer

Abstract

This project aims to fine-tune an existing large language model (LLM) for the task of generating concise and relevant traffic news reports. The goal is to automate the production of traffic updates for radio broadcast, based on structured data from the promet.si portal, a task currently being done manually by students at RTV Slovenija. We use various prompt engineering techniques and evaluate the effectiveness of outputs with (semi) automatic and human evaluation metrics. We take the following into account; accurate road naming, filtering of relevant events, correct filtering, text lengths and words and editorial guidelines. The report also discusses the preparation of datasets, challenges of generation in a low-resource language, and the design of an interface for interactive testing of model outputs.

Keywords

prompt engineering, text generation, traffic news, fine-tuning, data-to-text generation, RTV Slovenija

Advisors: Slavko Žitnik

Introduction

In an effort to modernize and automate the generation of traffic news for RTV Slovenija, this project explores the application of natural language processing (NLP) techniques for the automatic generation of Slovenian traffic reports. The main goal is to use a large language model (LLM), fine-tune it for this specific use case and to use prompt engineering for producing short and relevant traffic updates based on structured data from the promet.si portal. These reports are intended for radio presenters who regularly read them to inform the public.

The task is currently being done manually. While functional, this approach is more labor-intensive, prone to variability, and limited in terms of scalability. By automating the process, we hope to improve efficiency, consistency, and content quality, while following the editorial standards for language style, relevance, and format.

One of the things we focus on is the correct identification of relevant traffic events, accurate road naming, and appropriate text length and filtering. We achieve this with prompt engineering and parameter-efficient fine-tuning (e.g., LoRA). We evaluate generated text using automatic (precision, recall, F1) as well as human evaluation metrics. The project also touches on the design of interactive tools for testing model outputs, and explores best practices in dataset preparation and model evaluation.

In summary, this project aims to deliver a reliable and efficient system for automated traffic news generation by leveraging Large Language Models (LLMs), their fine-tuning, and prompt engineering techniques.

Related work

Generating informative traffic new from structured data falls within the broader domain of data-to-text (D2T) generation, specifically under table-to-text generation. For our project, structured traffic data is supplied by the promet.si portal in CSV format. The data includes fields such as road names, event types, timestamps and locations. The objective is to transform these data entries into concise and reliable reports that are ready for radio broadcasters to read.

Recent research has demonstrated that LLMs, when combined with data cleaning and model fine-tuning, are highly effective for table-to-text generation. For example, the HeLM (Highlighted Evidence augmented Language Model) framework [1] proposes a two-step approach, where a highlighter selects more important rows. The highlighter is then followed by a summarizer that produces human-readable text.

This method achieved state-of-the-art BLUE (bilingual evaluation understudy) and ROGUE (Recall-Oriented Understudy for Gisting Evaluation) scores, confirming that pre-selecting important data improves both model performance and text understandability.

Using LLMs in low resource environments requires parameter-efficient fine-tuning. Methods like LoRA (Low Rank Adaptation) [2] and QLoRA (Quantized Low Rank Adaptation) [3] allow fine-tuning models on modest hardware while maintaining high accuracy. Researchers have also found that smaller, fine-tuned models ensure reliability and resource efficiency [4].

Another essential part of our project is exploring and testing different prompt engineering techniques. This involves carefully crafting prompts to elicit specific, high-quality outputs from LLMs. Many studies have been done on prompt engineering techniques, in which researches test different prompt engineering techniques (zero-shot and few-shot prompting, chain-of-thought reasoning, and self-consistency decoding) [5]. In traffic news generation, these strategies help guide the model, while also helping it avoid unnecessary or irrelevant information.

Practical guidance for prompt engineering is provided by Prompting Guide AI (https://www.promptingguide.ai/), which offers extensive explanations and specific advice on different prompt engineering approaches. In addition, Ollama has demonstrated effective prompt workflow with models like Code Llama, especially for tasks involving structure data transformation. They offer different examples on how to prompt Code Llama to optimize relevance and clarity (https://ollama.com/blog/how-to-prompt-code-llama).

In conclusion, our project builds upon this body of research in table-to-text generation, fine-tuning methods, and prompt engineering. By integrating structured data processing, efficient LLM adaptation, and strategic prompt design, we aim to deliver automated and reliable compliant traffic news that meets the standards of RTV Slovenija for relevance, clarity, and factual accuracy.

Methods

The simplest way to tackle this challenge is through prompt engineering, which has been shown to significantly enhance the performance of LLMs. To improve the quality of our traffic reports, we will leverage various prompt engineering techniques such as few-shot prompting, which provides examples to guide the model's responses; chain-of-thought prompting, which encourages step-by-step reasoning for better accuracy; and prompt chaining, where multiple prompts are linked together to refine and build upon previous outputs. These techniques help ensure more accurate, detailed, and contextually relevant traffic reports.

0.1 Zero-shot prompting

Zero-shot prompting is a technique where we provide the model with a task description and ask it to generate a response without any examples. This strategy relies on the model being adequately trained and being exposed to large amounts of data, which allows it to generalize and understand the task at hand.

Few-shot prompting

Few-shot prompting involves providing the model with a few examples of the desired input-output format before asking it to generate a response. It helps the model understand the pattern of the expected output.

For example, we want to classify statements as **Positive**, **Negative**, or **Neutral**. We provide the LLM with the required task and some examples:

```
Task: Classify sentences as Positive,
      Negative, or Neutral.
Example 1: 'I hate being in Thailand.'
      - Negative,
Example 2: 'I like it when my sister
      visits.' - Positive
```

Then we provide a new sentence that we want to classify:

```
Now classify the following sentence: 'I
      enjoy long walks.'
```

The LLM is more likely to classify the sentence correctly, since it has been trained by examples. This method is very useful for classification, translation, and text generation.

Chain-of-thought prompting (CoT)

Chain-of-thought prompting is a technique that encourages the model to break down its reasoning process step by step. By asking the model to explain its thought process, we provide an intermediate reasoning step to guide model's responses, potentially leading to better results.

0.2 Prompt chaining

Prompt chaining is a technique where we link multiple prompts together to refine and build upon the model's previous outputs. This approach allows us to create a more structured and coherent response by guiding the model through a series of related tasks or questions. In our case we can first ask the model to generate a list of relevant traffic events, and then ask it to summarize those events into a report.

0.3 Self-consistency prompting

Self-consistency prompting involves generating multiple outputs for the same prompt and selecting the most appropriate one. This helps reduce variability in model responses and improves reliability. It is especially useful with models like GaMS 9B, which may produce inconsistent results across runs.

0.4 LLM fine-tuning

Another way we can tackle this problem is by fine-tuning an existing LLM model. This approach allows us to adapt the model to our specific task and domain, in our case generating traffic reports, by training it on a smaller dataset of

Slovenian traffic reports, helping the model learn the specific language patterns, terminology, and style used in this context. We will use the LoRA (Low-Rank Adaptation) method, which is a parameter-efficient fine-tuning technique. It works by injecting a low-rank trainable matrix into each layer of the pre-trained model, allowing us to fine-tune only a small number of parameters while keeping the rest of the model frozen. Empirical studies have shown that LoRA can achieve comparable performance to full fine-tuning while significantly reducing the number of trainable parameters and computational resources required.

0.5 Data preparation

The data we will use for training and evaluation is sourced from the promet.si portal, which is the main source of traffic information in Slovenia, and will be used as input data and also RTV Slovenija's traffic news archive and formatting guidelines, which will server as expected output data and guidelines for prompting.

Evaluation

To evaluate the performance of the different approaches, we will use a combination of automatic and human evaluation metrics. For automated evaluation, we decided to use BLEU (bilingual evaluation understudy), which compares the machine-generated text to one or more reference texts and produces a score between 0 and 1, with values closer to 1 representing more similar texts. Another automatic evaluation metric we will be using is BERTScore, which uses BERT embeddings to measure the semantic similarity between generated and reference texts. We will also use ROUGE-L, which evaluates the longest common subsequence between generated and reference outputs. Additionally, we will use manual evaluation, where human evaluators will assess the generated traffic reports based on criteria such as relevance, clarity, and adherence to editorial guidelines. We will use automatic evaluation metrics on all generated reports, and then select a small subset of the reports for human evaluation, to better understand the correlation between the evaluation methods.

References

[1] Junyi Bian, Xiaolei Qin, Wuhe Zou, Mengzuo Huang, Congyi Luo, Ke Zhang, and Weidong Zhang. Helm: Highlighted evidence augmented language model for enhanced table-to-text generation, 2024.

[2] Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation, 2024.

[3] Yelysei Bondarenko, Riccardo Del Chiaro, and Markus Nagel. Low-rank quantization-aware training for llms, 2024.

[4] Joy Mahapatra and Utpal Garain. Impact of model size on fine-tuned llm performance in data-to-text generation: A state-of-the-art investigation, 2024.

[5] Shubham Vatsal and Harsh Dubey. A survey of prompt engineering methods in large language models for different nlp tasks, 2024.