CMPSC 497 Final Project Report

Parth Gosar, Sahil Pardasani

Date: April 25, 2025

1. Problem Overview

The project objective was to build a system that generates detailed research methodologies from environmental problem statements. We aimed to demonstrate practical fine-tuning of open-source models, moving beyond zero-shot prompting, and exploring natural language generation (NLG) tasks in scientific domains like climate science.

2. Dataset Construction

We used the ClimateFever dataset from Hugging Face, originally designed for claim verification tasks in climate science. It consists of claims and supporting evidence sentences. For our task, we adapted the claims as 'Problem Statements' and synthesized evidence into structured 'Research Approaches'. We manually filtered for relevance using climate-related keywords like sustainability, renewables, and biodiversity. The final dataset contained approximately 500 high-quality (problem, approach) pairs.

3. LLM Selection and Training

We fine-tuned two models: T5-small and facebook/bart-base. T5-small was selected for lightweight, quick prototyping, while BART was selected for richer semantic generation. Tokenization involved input sequences limited to 64 tokens and output sequences to 256 tokens. We utilized Hugging Face Trainer API with GPU acceleration on Google Colab Pro (Tesla T4 GPU).

4. Hyperparameter Tuning

We employed Optuna for hyperparameter tuning. The search space included learning rates, batch sizes, and number of epochs. The best hyperparameters were found to be:
- Learning Rate: 5e-5
- Batch Size: 8
- Epochs: 4

5. Evaluation and Experiments

We evaluated the models using ROUGE-L and BERTScore metrics. Here are the results:

Model	Dataset Size	ROUGE-L	BERTScore
T5-small	~100	0.421	0.824
facebook/bart-base	~500	0.2068	0.8510

Manual analysis revealed that BART provided superior semantic richness and coherence compared to T5, though T5 outputs were shorter and structurally consistent.

6. Model Comparison and Recommendations

T5-small is ideal for scenarios requiring quick fine-tuning and interpretable outputs. BART, however, delivers higher semantic fidelity and more complex text generation, making it better suited for scientific documentation tasks.

7. Thoughts and Reflections

Key challenges included computational resource constraints, managing long training times for BART, and designing effective keyword filtering for dataset preparation. Lessons learned: the value of semantic-focused evaluation (BERTScore) over pure n-gram matching, and the importance of robust hyperparameter tuning using automated search.

8. Conclusion and Future Work

Future improvements involve using controlled generation techniques (like keyword-conditioned decoding), trying distilled versions of BART for faster inference, and expanding to other scientific disciplines beyond environmental science.

9. GitHub Repository

GitHub Repository: https://github.com/psg0009/CMPSC-497-Final-Project-

Appendix A. Sample (Problem, Approach) Pair

Problem: Oceans are acidifying due to rising CO2 levels.
Approach: Conduct long-term pH monitoring, assess coral resilience, correlate acidification trends with regional CO2 fluxes.

Appendix B. Code Snippet (Training with BART)

```
from transformers import BartTokenizer, BartForConditionalGeneration
model = BartForConditionalGeneration.from_pretrained('facebook/bart-base')
tokenizer = BartTokenizer.from_pretrained('facebook/bart-base')

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

# TrainingArguments
training_args = TrainingArguments(output_dir='./results', num_train_epochs=4,
per_device_train_batch_size=8, learning_rate=5e-5, evaluation_strategy='epoch',
save_strategy='epoch')

# Trainer setup
trainer = Trainer(model=model, args=training_args, train_dataset=train_dataset,
eval_dataset=eval_dataset, tokenizer=tokenizer)
trainer.train()
```