Project 3

Mohammed Musthafa Rafi
mohd7@iastate.edu

Abstract

This report presents a comprehensive approach to scientific paper title generation from abstracts. Three distinct methods were developed and evaluated: a rule-based approach using key term extraction and pattern recognition, a neural approach utilizing the fine-tuned BART model, and a hybrid approach combining both techniques with a quality assessment framework. Evaluation on 1,097 scientific papers from the QTL dataset demonstrates that neural and hybrid methods substantially outperform rule-based approaches, with the hybrid method achieving the highest scores (BLEU: 0.1297, ROUGE-2: 0.2616, ROUGE-L: 0.4209). Error analysis reveals challenges in capturing central contributions and specialized terminology, suggesting avenues for future improvement.

1 Method

I implemented a hybrid approach for scientific paper title generation from abstracts, combining a pre-trained language model with rule-based components. My implementation consists of three main approaches:

Approach 1: Rule-Based Title Generation.

1. **Preprocessing**: Extract key scientific terms, method keywords, species mentions, and biomedical terminology from abstracts.
2. **Pattern Identification**: Identify common structural patterns in scientific titles (e.g., "Analysis of X in Y").
3. **Title Generation**: Generate titles using templates based on extracted information, prioritizing the first sentence of the abstract for main topic identification.
4. **Evaluation**: Compute BLEU, ROUGE-2, and ROUGE-L against ground truth titles.

Approach 2: BART Model Title Generation.

1. **Preprocessing**: Clean and tokenize abstract text using the BART tokenizer with maximum sequence length of 512 tokens. Abstracts longer than this limit are truncated.
2. **Model Architecture**: Fine-tune the pre-trained `facebook/bart-base` model (139M parameters), which uses a bidirectional encoder and autoregressive decoder architecture with 12 transformer layers in both components.
3. **Training Configuration**: Train for 3 epochs with AdamW optimizer (learning rate 5e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay=0.01), batch size of 4, gradient accumulation over 2 steps, and linear learning rate warmup over 500 steps.
4. **Input Representation**: Raw abstracts are directly encoded through the BART encoder, with no special prompt prefix, using BART's standard positional embeddings and byte-pair encoding.
5. **Generation Strategy**: Generate titles using beam search (beam size=4) with length penalty of 2.0, maximum generation length of 64 tokens, no-repeat n-gram size of 2, and early stopping when all beam hypotheses finish.
6. **Computational Resources**: Training performed on a single NVIDIA GPU with 16GB VRAM, taking approximately 4 hours for complete fine-tuning.

Approach 3: Hybrid Title Generation.

1. **Multiple Candidates**: Generate title candidates using both the rule-based approach and fine-tuned BART model.
2. **Textual Analysis**: Automatically extract domain-specific terminology from each abstract using regex-based patterns and frequency analysis. Key scientific terms are identified through capitalization patterns, biomedical lexicons, and n-gram frequency analysis.
3. **Quality Assessment**: Implement a scoring function that evaluates candidates based on: (1) scientific term coverage (40%), (2) linguistic structure conformity to scientific conventions (30%), and (3) title length and conciseness

(30%).

4. **Selection Algorithm**: Use a weighted decision function that selects the BART-generated title when it contains at least 60% of the key scientific terms identified in the abstract and has length between 4-15 words; otherwise fall back to the rule-based title.
5. **Post-processing**: Apply standardization to ensure consistent capitalization of scientific terms and remove any generation artifacts (e.g., repetitive phrases, truncated words).

2 Results

I evaluated all three approaches on a test set of 1,097 scientific papers from the QTL dataset. Table 1 summarizes the performance metrics for each approach.

Approach	BLEU	ROUGE-2	ROUGE-L
Rule-Based	0.0147	0.0399	0.1770
BART Model	0.1293	0.2616	0.4205
Hybrid	0.1297	0.2616	0.4209

Table 1: Performance comparison of title generation approaches on test set.

The results show that both the BART model and hybrid approaches substantially outperform the rule-based approach across all metrics. The hybrid approach achieves marginally better scores than the pure BART approach, suggesting that incorporating rule-based knowledge can complement neural generation in specific cases.

3 Discussion

The results demonstrate several interesting findings about scientific title generation:

Neural vs. Rule-Based Approaches. The significant performance gap between the rule-based approach and the neural approaches (BLEU: 0.0147 vs. 0.1293/0.1297) underscores the importance of contextual understanding and domain knowledge embedded in pre-trained language models. The rule-based approach, despite using scientific term extraction and structural patterns, lacks the semantic understanding required for accurate title generation.

Hybrid Advantage. The hybrid approach's slight improvement over pure BART (BLEU: 0.1297 vs. 0.1293) suggests that rule-based selection criteria can help in specific cases where the neural model might generate less precise titles. However, the minimal difference indicates that the neural model alone is already quite effective for most papers.

Metric Analysis. While BLEU scores are relatively low (0.1297), which is typical for generation tasks, the ROUGE-L score of 0.4209 indicates that the generated titles capture a significant portion of the longest common subsequence with the reference titles. This suggests the models are identifying key concepts but may express them differently than the original authors.

Output Analysis. The comprehensive evaluation generated several output files for detailed analysis:

- `generated_titles_comparison.csv`: Contains all titles from all three approaches
- `rule_based_titles.csv`, `bart_titles.csv`, `hybrid_titles.csv`: Individual approach outputs

Analysis of these files reveals that rule-based titles tend to be formulaic and often miss the central contribution, while BART-generated titles capture the paper's main focus more accurately but occasionally include repetitive phrases. Hybrid titles maintain BART's semantic accuracy while improving specific terminology coverage.

The results demonstrate that while automatic title generation remains challenging, neural approaches significantly outperform rule-based methods. The slight advantage of the hybrid approach suggests that there is value in combining neural generation with domain-specific knowledge, particularly for capturing technical terminology in scientific papers.

References

[1] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[2] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

[3] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.

A Appendix: Resources

Below are relevant resources and references used in this project:

- https://github.com/itsMustafamr/Project3_COMS_579 (GitHub repository with all project code)
- https://huggingface.co/docs/transformers/model_doc/bart (Hugging Face BART documentation)
- https://huggingface.co/spaces/evaluate-metric/bleu (BLEU score reference)
- https://huggingface.co/spaces/evaluate-metric/rouge (ROUGE metrics reference)
- https://scikit-learn.org/stable/ (Scikit-learn for data processing utilities)
- https://www.nltk.org/ (NLTK for text processing)

All Python code, model training scripts, and evaluation results are available in the GitHub repository.