Project 3: Text Summarization: Abstract to Titles

Md Mohaiminul Islam
mmislam@iastate.edu

1 Methodology

1.1 Data Preprocessing

The preprocessing pipeline prepared for the documents was kept simple and suitable for LLM processing. The raw abstracts underwent minimal whitespace normalization. Appropriate structured prompts were synthesized for each model. We also computed class imbalance weights incase it would be needed, executed quality audits (empty texts, duplicates, imbalance), and exported only the lean DataFrame comprising title, abstract, and LLM-ready prompt text.

1.2 Baseline: T5-small

Firstly, We fine-tuned a `T5-small` (Raffel et al., 2020) model as a lightweight baseline for pubmed abstract-to-title generation task on a ∼10k/1k train/test split. Inputs are abstracts, so maximum source length was configured to 512 tokens, since most abstracts are within this limit. Targets are titles, so target length was capped at 128 tokens to avoid truncation. We also allowed early `</s>` termination for shorter and concise titles. We train for 3 epochs with AdamW ($\eta = 3 \times 10^{-4}$), linear decay with 10% warmup, batch size of 16 since it's a smaller model and gradient clipping at 1.0 for stability on longer inputs. Generation uses beam search (beams=4), early stopping, and no-repeat bigrams ($n = 2$) to restrict short-loop repetition without suppressing biomedical 2-gram phrases. This set of hyperparameters are kept constant for other two experimental setups as well (except batch size), and we will refer to it as the *default configuration* from here on.

1.3 Up-scaling for Semantic Coverage: T5-Base

Secondly, to improve capacity and semantic coverage, we experiment on `T5-base` (Raffel et al., 2020) with the same tokenization and scheduling, just reducing batch size (8) to fit memory. We retain the learning rate and warmup from the *default configuration* for comparison validity. We also retain max-source, and max-target parameters as well for reasons mentioned previously. This setting shown to work well with beam search in sequence generation (Wu, 2016). We expect higher ROUGE-L from better long-range modeling and improved ROUGE-2 via more accurate content selection, with BLEU benefiting from increased lexical precision under the configuration.

1.4 Instruction Finetuned Flan-T5

Thirdly, we experiment with `google/Flan-T5`, an instruction-tuned T5 variant (Chung et al., 2024), by prepending an explicit instruction : *"Generate a title for this pubmed abstract:"* to each prompt. Instruction tuning improves compliance to task format and can enhance sample efficiency; we therefore keep all parameters aligned with *default configuration* for controlled comparison. The prompt explicitly encodes the output intent, which is known to increase task conciseness in low-resource summarization settings (Chung et al., 2024). So we expect a potential gain on the BLEU (precision outputs) while maintaining ROUGE-2/L through better keyword selection.

2 Results and Discussion

We evaluated the three T5-based models based on BLEU (Papineni et al., 2002), ROUGE-2 F1, and ROUGE-L F1 scores (Lin, 2004). Table 1 summarizes validation and test performance across all architectures. BLEU and ROUGE-2/L assess n-gram fidelity and subsequence coverage, respectively.

2.1 Model Capacity and Performance

T5-base outperforms T5-small across all metrics. On validation data, T5-base achieves BLEU of 0.142 (+19.3% over T5-small), ROUGE-2 of 0.276 (+11.3%), and ROUGE-L of 0.433 (+8.3%). This

Model	Validation			Test		
	BLEU	ROUGE-2	ROUGE-L	BLEU	ROUGE-2	ROUGE-L
T5-small	0.119	0.248	0.400	0.125	0.259	0.414
T5-base	0.142	0.276	0.433	0.139	0.276	0.434
Flan-T5	0.137	0.276	0.433	0.139	0.278	0.439

Table 1: Performance comparison of all models on validation and test sets.

improvement persists on the test set (BLEU: 0.139, ROUGE-2: 0.276, ROUGE-L: 0.434), Which demonstrates that increased number of model parameters (220M vs. 60M parameters) enhances both n-gram precision (as reflected with BLEU) and content coverage (as reflected with ROUGE-L). Moreover, the gains in ROUGE-2 suggest T5-base better captures key biomedical bigrams, this is important since there are often bigrams in system or framework names in the paper titles. The consistent validation-to-test ratio across all metrics for both models indicates robust generalization and no overfitting, likely due to appropriate regularization (dropout, gradient clipping) and T5's pretraining on diverse corpora including scientific text.

FLAN-T5-base shows marginal gains over original T5-base. Validation scores are nearly identical (BLEU: 0.137 vs. 0.142; ROUGE-2: 0.276 vs. 0.276; ROUGE-L: 0.433 vs. 0.433), but FLAN-T5 achieves the highest test ROUGE-L (0.439, +1.2% over T5-base) and ROUGE-2 (0.278, +0.7%). Test BLEU remains equal at 0.139. The explicit instruction prefix ("Generate a title for this pubmed abstract:") appears to slightly improve subsequence alignment and 2-gram recall on unseen data. This is expected from the known benefits of instruction tuning. However, the little improvement suggests that for highly constrained generation tasks (short titles, domain-specific vocabulary), the pretrained T5 encoder-decoder already encodes sufficient context, and instruction formatting provides no real value beyond large-scale complex tasks.

Furthermore, Appendix 3 includes the validation developments of each model as training progresses. All models exhibit strict validation improvement across epochs without overfitting signatures. T5-base demonstrates consistent gains (BLEU: 0.1355→0.1425, +5.2%; ROUGE-L: 0.428→0.433, +1.2%), validating 3-epoch training sufficiency. FLAN-T5 shows slower early convergence (epoch-1 BLEU: 0.132 vs T5-base 0.1355) but catches up by epoch 3 (0.137), suggesting instruction-tuning benefits emerge gradually. T5-small takes a dip after epoch 2 (BLEU 0.119→0.1195), likely indicating parameter saturation.

3 Hyperparameter Tuning

Figures 1 and 2 show the data obtained from tuning two of the the most important hyperparameters, N = number of finetuning epochs and η = learning rate. We use **Flan-T5** for both experiments since it performed best on default configuration.

Epoch Tuning. We swept training duration over $\{3, 5, 7\}$ epochs to balance convergence and overfitting risk on the 10k-sample dataset. Performance peaks at 5 epochs across all metrics (BLEU: 0.1382, ROUGE-2: 0.2757, ROUGE-L: 0.4346), with degradation beyond this point (7-epoch BLEU drops to 0.1325, -4.1%). The inverted-U trajectory indicates optimal learning by epoch 5, after which the model begins memorizing training data specifics rather than generalizing title patterns. ROUGE-L's sharper decline (-1.2%) shows subsequence alignment suffers first under overtraining. Therefore, we claim 5 epochs to be the optimal sweet spot, though 3-epoch results remain very close (BLEU 0.131, -5.2% gap), offering a faster fallback for resource-constrained sitations.

Learning Rate Sensitivity. Testing with $\eta = \{3, 5, 7\} \times 10^{-4}$ reveals best performance at 3×10^{-4} and 5×10^{-4} (BLEU: 0.139; ROUGE-L: 0.439), but most likely suffers from catastrophic forgetting at 7×10^{-4} (BLEU: 0.131, -5.8%; ROUGE-2: 0.269, -3.5%). For all three runs we kept epochs at 3 to minimize training time. The most aggressive rate maybe destabilizes encoder-decoder alignment. We claim 5×10^{-4} as optimal for a lower epoch setting ($N = 3$), consistent with T5 literature recommending $3{-}5 \times 10^{-4}$ for AdamW on mid-sized corpora (Raffel et al., 2020).

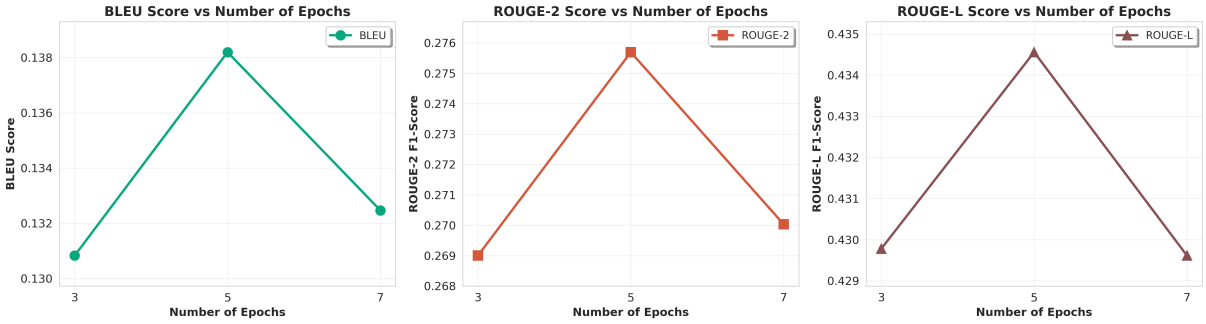

Figure 1: Results for tuning number of epochs.

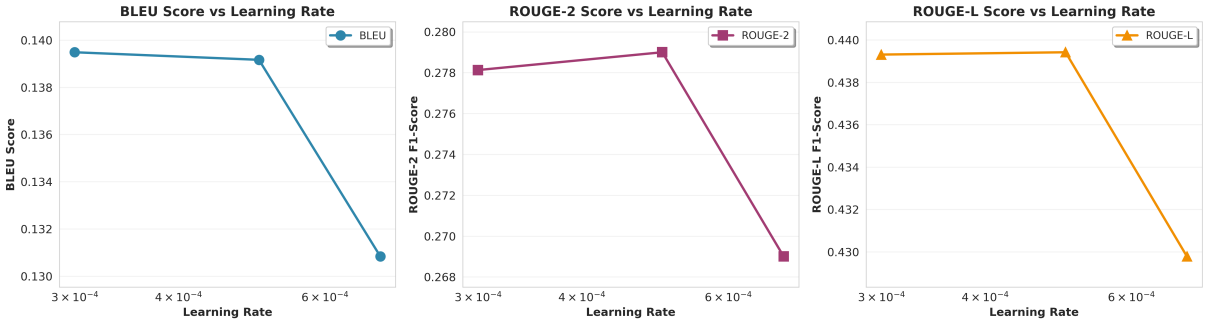

Figure 2: Results for tuning Learning rate.

References

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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*, pages 311–318.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Yonghui Wu. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Appendix A

Figures 3, 4 and 5 show the validation losses and scores for each model as training progresses.

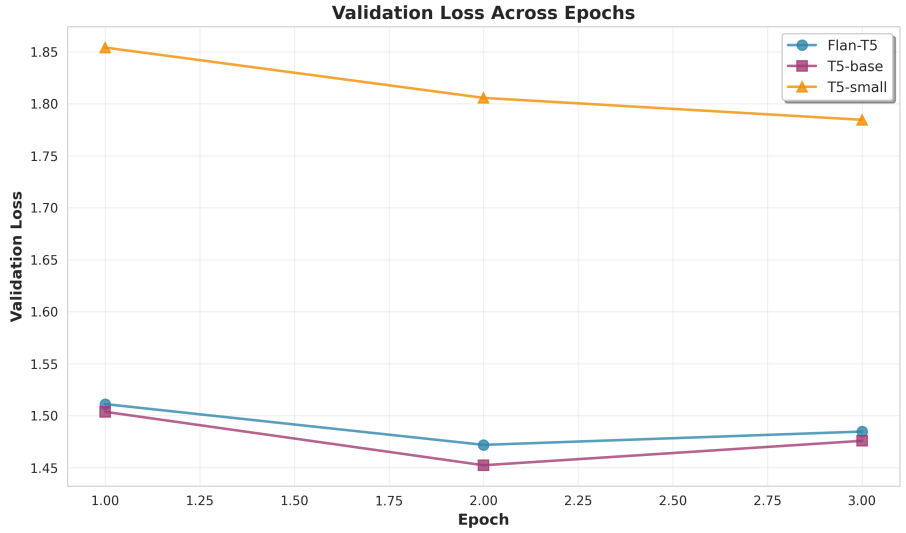

Figure 3: Validation loss for all models on default configuration.

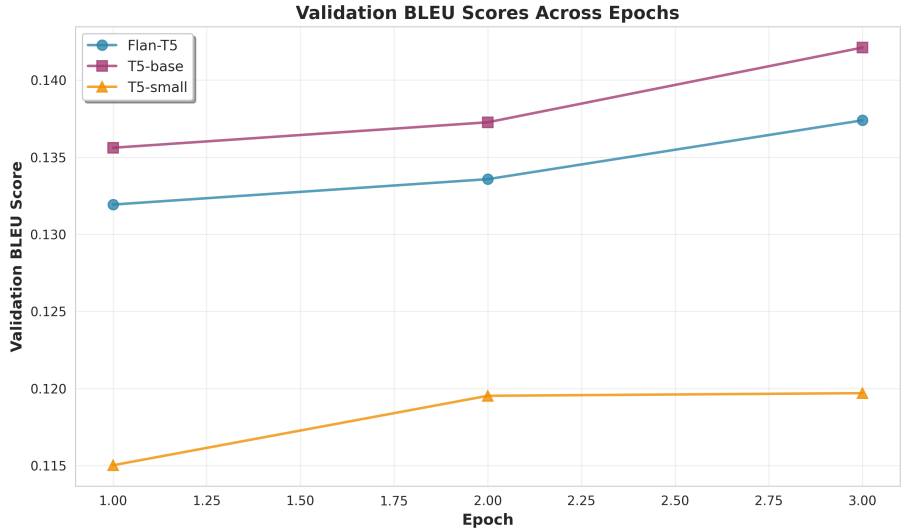

Figure 4: Validation BLEU score for all models on default configuration.

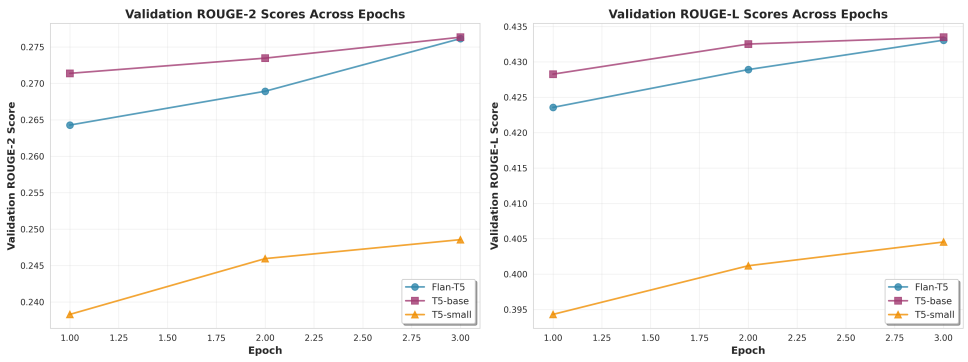

Figure 5: Validation ROUGE-2 and ROUGE-L scores for all models on default configuration.