Claim Normalization: BART and T5

NLP Assignment 3 | Task 2 | Gour Krishna Dey | MT24035

1. Introduction and Problem Definition

In the age of rapid information dissemination across social media platforms, ensuring the factual accuracy of online claims has become a growing challenge. Social media posts often contain informal, unstructured, and noisy text, making them difficult for fact-checking systems to process directly. To address this, the task of **Claim Normalization** has emerged as a crucial step in the fact-checking pipeline.

Claim Normalization involves transforming informal social media posts into **clear, structured, and factual statements**, also known as *normalized claims*. This process allows downstream fact-checking tools to better assess the veracity of claims by working with concise and coherent representations.

Our approach is inspired by the paper *"From Chaos to Clarity: Claim Normalization to Empower Fact-Checking"* which formalizes this task and provides a benchmark dataset known as **CLAN** (Claim Level Annotated News). The CLAN dataset comprises real-world social media posts, each paired with professionally normalized claims.

To begin our work, we first designed and implemented a complete preprocessing pipeline for the CLAN dataset to make it suitable for fine-tuning transformer-based text generation models—specifically **BART-base** and **T5-small**.

2. Preprocessing Pipeline

Effective preprocessing is essential in claim normalization, especially since social media posts are highly noisy and inconsistent. Our pipeline includes the following steps:

1. **Contraction & Abbreviation Expansion**:
   We expanded common contractions (e.g., *he'll → he will*, *she's → she is*) and domain-specific abbreviations (*Gov. → Governor*, *Feb. → February*, *VP → Vice President*, etc.). This ensures the generated claims are formal and grammatically correct.

2. **Text Cleaning**:
   All posts were cleaned by:

   – Removing URLs, mentions (@user), and hashtags.
   – Removing non-alphanumeric characters except basic punctuation.
   – Eliminating extra whitespace and converting text to lowercase.
   – Stripping emojis and special tokens that may mislead the model.

3. **Data Splitting**:
   The dataset was split into **70% training**, **15% validation**, and **15% test** sets to maintain a balanced evaluation setup.

The result of preprocessing was a clean and well-structured dataset, ready for fine-tuning on generation tasks using transformer models.

3. BART-Base Pipeline

BART-base was used as a sequence-to-sequence model, where noisy social media posts were passed as input and the normalized claims as target sequences. The model was fine-tuned using cross-entropy loss, with evaluation carried out at regular steps using ROUGE-L, BLEU-4, and BERTScore metric

Key Hyperparameters:

- Learning Rate: 3e-5
- Batch Size: 8
- Epochs: 5
- Evaluation Strategy: Validation every 200 steps
- Optimizer: AdamW

BART-BASE: Epoch-wise Training Summary

Epoch	Train Loss	Val Loss	ROUGE-L	BLEU-4	BERTScore
1	2.0445	0.4986	0.3368	0.1911	0.8807
2	0.4888	0.4822	0.3351	0.1886	0.8802
3	0.4124	0.4738	0.3380	0.1913	0.8809
4	0.3522	0.4775	0.3356	0.2066	0.8803
5	0.2990	0.4912	0.3310	0.2139	0.8819

The model showed a consistent decrease in training loss and a relatively stable validation loss, with improved BLEU-4 and BERTScore across epochs.

4. T5-Small Pipeline

T5-small was employed in a similar manner, with the input text prefixed by a task-specific instruction (e.g., `"normalize: <post>"`) as per the T5 paradigm. This enabled the model to learn the transformation task explicitly. Due to limited GPU resources, the smaller T5 variant was chosen to balance performance and compute efficiency.

Key Hyperparameters:

- Learning Rate: 3e-5
- Batch Size: 8
- Epochs: 5
- Input Prefix: `"normalize: "` added to every input text
- Same evaluation setup as BART

T5-SMALL: Epoch-wise Training Summary

Epoch	Train Loss	Val Loss	ROUGE-L	BLEU-4	BERTScore
1	3.2131	2.9105	0.3127	0.1422	0.8742
2	2.9375	2.8050	0.3154	0.1581	0.8751
3	2.8765	2.7602	0.3179	0.1659	0.8760
4	2.8211	2.7399	0.3210	0.1704	0.8767
5	2.8024	2.7329	0.3195	0.1687	0.8771

While T5-small demonstrated steady improvement across all metrics, its performance plateaued below BART-base on the validation set.

5. Results on Validation Set

Model Evaluation Metrics (on Validation Set)

Model	BLEU-4 Score	ROUGE-L Score	BERTScore (F1)
BART-base	22.20	33.14	88.22
T5-small	17.31	35.64	87.15

Both models performed reasonably well, but **BART-base consistently outperformed T5-small in terms of BLEU-4 and BERTScore**, indicating better fluency and semantic alignment in the generated claims.

6. Visualizations
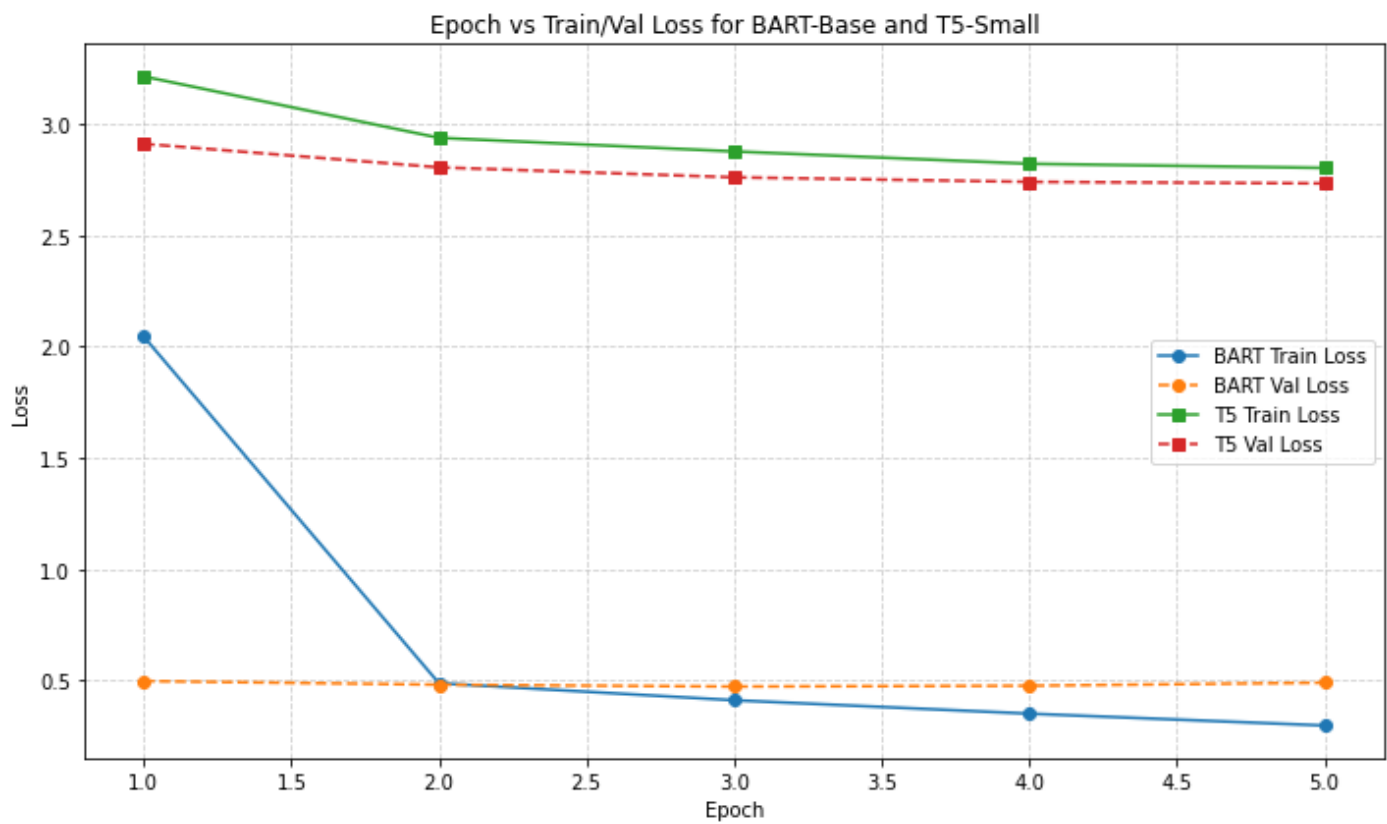
(i) Epoch vs Train and Validation Loss

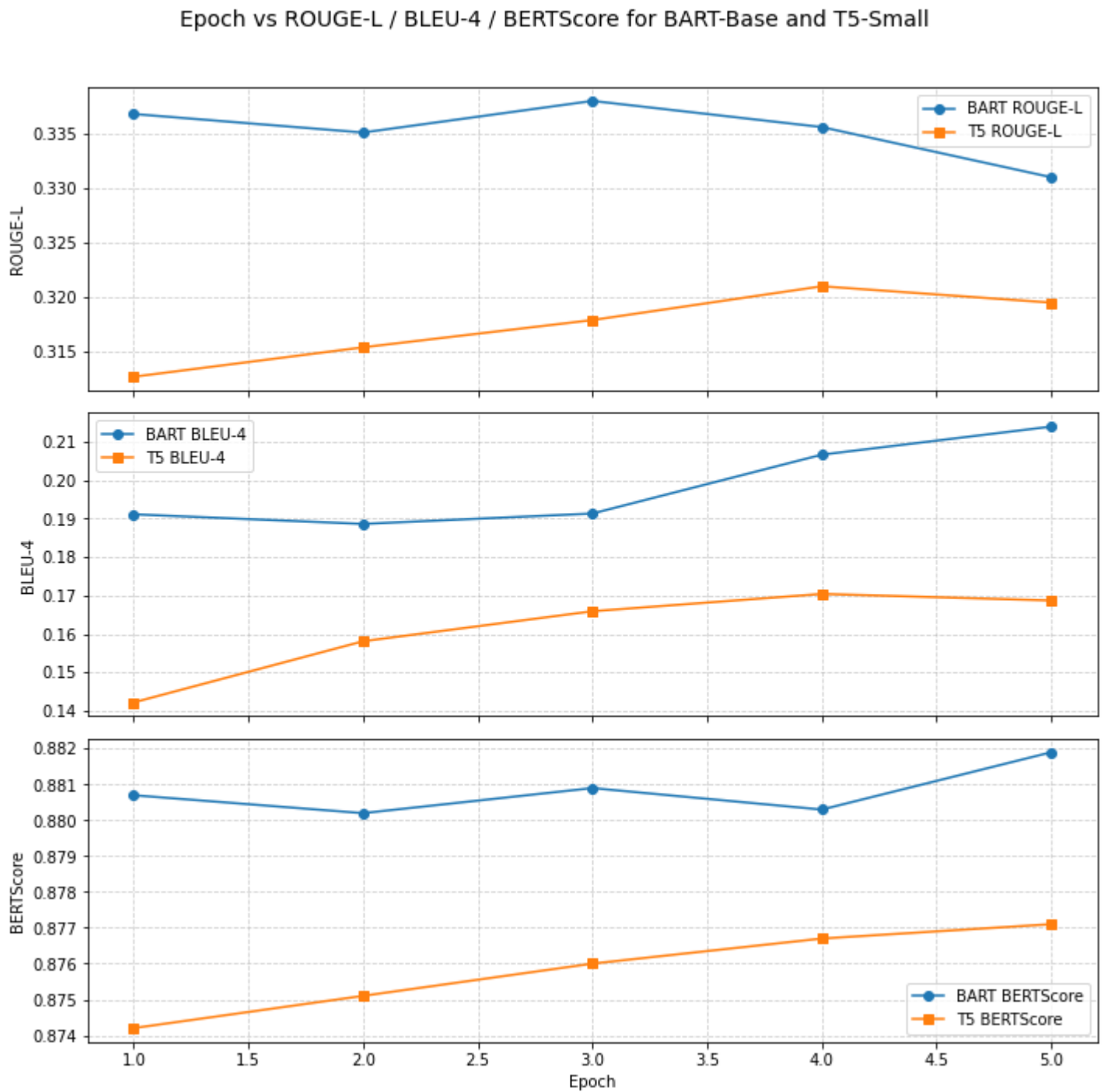*Fig.1.Graph shows declining training and validation loss over epochs for both models.*

Fig.2.Graph shows BLEU-4, ROUGE-L & BERTScore trends for both BART and T5 over epochs.

7. Discussion and Analysis

Both BART-base and T5-small showed the ability to perform the claim normalization task effectively. However, BART-base emerged as the better model for the following reasons:

- **Stronger Generation Backbone**: BART uses a denoising autoencoder architecture, which makes it robust at reconstructing cleaned, well-formed outputs from noisy inputs.
- **Higher Semantic Fidelity**: BERTScore and BLEU-4 metrics suggest BART's outputs were closer in meaning and structure to ground-truth normalized claims.
- **Stability in Training**: BART showed more stable validation losses, indicating less overfitting and better generalization.

Given GPU memory and runtime limitations, T5-small offered a more lightweight alternative, but BART-base was selected as the final model due to its superior performance in high-fidelity claim rewriting.

8. Conclusion

This project demonstrates that **claim normalization** is a crucial preprocessing step for downstream fact-checking tasks. By fine-tuning transformer-based models on the *CLAN dataset*, we show that **BART-base** achieves strong performance in transforming informal social media claims into normalized, verifiable statements.

In resource-constrained environments like Google Colab, smaller models such as **T5-small** can still yield competitive results, albeit with minor trade-offs in accuracy. Overall, our results validate the importance of architectural choices and preprocessing in building effective claim normalization pipelines.

---------- X ----------