Supplementary Materials

Classification of Indonesian Language News Documents Using RNN and Transformers

This supplementary material presents detailed transformer model architectures, training parameters, and comprehensive evaluation metrics to complement our comparison of RNN and transformer models for Indonesian news classification. Our analysis provides deeper insights into why transformer models outperform RNN approaches despite their larger parameter counts.

A. Transformer Model Architectures and Mathematical Formulations

The transformer models implemented in our study utilize the standard transformer architecture with multi-head self-attention mechanisms [1]. This architecture processes tokens in parallel rather than sequentially, enabling more efficient contextual relationship modeling. All three transformer variants in our study (IndoBERT, XLM-RoBERTa, and multilingual BERT) share a common architecture of 12 transformer layers, 768 hidden units, and 12 attention heads, differing primarily in their pre-training data sources and tokenization strategies [2], [3].

The core self-attention mechanism is expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

$$\text{MultiHead}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2}$$

$$\text{head}_i = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V) \tag{3}$$

where Q, K, and V are query, key, and value projections derived from input embeddings. Each transformer layer includes a position-wise feed-forward network:

$$\text{FFN}(x) = \text{GELU}(xW_1 + b_1)W_2 + b_2 \tag{4}$$

These mathematical formulations illustrate why transformers can process information in parallel, compared to the sequential nature of RNNs. This leads to potential efficiency advantages when leveraging modern GPU hardware.

B. Training Hyperparameters

Our experimental approach employed consistent training configurations across all models to ensure fair comparison. Table I details the unified training hyperparameters used for all transformer models.

As shown in Table I, we used the AdamW optimizer with a learning rate of 2e-5 and a weight decay of 0.01, which proved effective for fine-tuning transformer models. We implemented a linear learning rate schedule with 10% warmup to stabilize the training process. Additionally, a batch size of 32 was chosen to balance computational efficiency with the quality of gradients. These carefully calibrated parameters allowed the transformer models to leverage their pre-training for the Indonesian news classification task effectively.

TABLE I
TRAINING HYPERPARAMETERS FOR INDONESIAN NEWS CLASSIFICATION

Hyperparameters	Config
optimizer	AdamW
learning rate	2e-5
batch size	32
LR schedule	linear with warmup
warmup ratio	0.1
training epochs	10
weight decay	0.01
dropout	0.15
gradient clipping	1.0
max sequence length	512

C. Transformer Model Implementation

We implemented a custom classification head on top of the pre-trained models for all transformer variants. Our implementation unfreezes all transformer layers during fine-tuning, which proved critical for achieving optimal performance. The following code snippet illustrates our approach:

```python
class TransformerClassifier(nn.Module):
    def __init__(self, pretrained_model, n_classes, dropout=0.15):
        super().__init__()
        # Load pre-trained transformer model
        self.transformer = BertModel.from_pretrained(pretrained_model)

        # Classification head with dropout regularization
        self.dropout = nn.Dropout(dropout)
        self.classifier = nn.Linear(
            self.transformer.config.hidden_size, n_classes)

    def forward(self, input_ids, attention_mask):
        # Forward pass through transformer
        outputs = self.transformer(
            input_ids=input_ids,
            attention_mask=attention_mask
        )

        # Use pooled output for classification
        pooled_output = outputs.pooler_output
        return self.classifier(self.dropout(pooled_output))
```

This implementation uses the [CLS] token representation (pooled output) from a pre-trained transformer for classification, incorporating dropout regularization to prevent overfitting. We discovered that enabling gradient flow through all layers of the transformer, instead of freezing the early layers, was crucial for achieving optimal performance in Indonesian text classification.

D. Computational Efficiency Analysis

A surprising finding of our research is the superior computational efficiency of transformer models despite their larger parameter counts. Table II presents processing efficiency metrics measured on RTX 3090 hardware.

As illustrated in Table II, multilingual BERT achieved the highest throughput, processing 21,228 tokens per second, despite having the largest number of parameters at 167.36 million. This efficiency advantage is notable, as it allows

TABLE II
PROCESSING EFFICIENCY AND TRAINING ACCURACY

Model	**Parameters**	**Tokens/s**	**Train Accuracy**
	(M)	(RTX 3090)	(%)
RNN [4]	1.37	11,863	92.00
XLM-RoBERTa	125.73	18,683	96.84
mBERT	167.36	21,228	96.42
IndoBERT	111.09	16,002	98.57

multilingual BERT to process text 1.79 times faster than the RNN model, which has significantly fewer parameters at 1.37 million.

The contrasting performance can be attributed to the transformer architecture, which effectively utilizes the parallel processing capabilities of modern GPUs. While RNN models are required to process tokens sequentially, transformers can process all tokens simultaneously. This parallel processing ability leads to a more efficient use of GPU resources, resulting in higher throughput.

This efficiency advantage can be quantified as a speedup factor S:

$$S = \frac{T_{transformer}}{T_{RNN}} = \frac{\text{tokens/s}_{transformer}}{\text{tokens/s}_{RNN}} \tag{5}$$

The speedup factors for IndoBERT, XLM-RoBERTa, and mBERT are 1.35, 1.57, and 1.79, respectively, even though these models have 81 to 122 times more parameters than the RNN model. These significant improvements in efficiency demonstrate that architectural advantages can outweigh differences in parameter counts when utilizing modern GPU hardware.

While it may not be the fastest model, IndoBERT achieved the highest training accuracy at 98.57%. This underscores the benefits of language-specific pre-training for Indonesian text classification. IndoBERT's accuracy represents a notable 6.57 percentage point improvement over the RNN model's accuracy of 92.00%, highlighting the importance of specialized pre-training for effectively capturing Indonesian linguistic patterns.

E. Training Dynamics Analysis

To gain deeper insights into the training dynamics of different architectures, we analyzed their performance across epochs, as demonstrated in Figure 1.

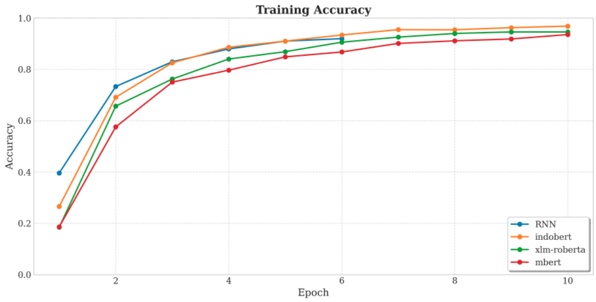

Fig. 1. Training accuracy curves across epochs

Figure 1 highlights the distinct learning trajectories of RNN and transformer architectures. Initially, the RNN model exhibits rapid learning, achieving approximately 92% accuracy within the first six epochs. However, it subsequently reaches a plateau, suggesting that it has reached its representational capacity early in the training process. In contrast, the transformer models demonstrate a more gradual and sustained enhancement in accuracy over the entire duration. Notably, IndoBERT consistently outperforms its transformer counterparts beginning from epoch five, while XLM-RoBERTa and mBERT follow similar, though slightly less pronounced, improvement curves. These observations indicate that transformer models have a superior ability to continually refine their understanding of the intricacies of Indonesian text, even in the later stages of training, unlike RNNs which saturate earlier.

REFERENCES

[1] S. Khan, M. Naseer, M. Hayat, S. Zamir, F. Khan, and M. Shah, "Transformers in vision: A survey," in *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 121–125, IEEE, 2022.

[2] B. Wilie, K. Vincentio, G. Winata, S. Cahyawijaya, X. Li, *et al.*, "Indonlu: Benchmark and resources for evaluating indonesian natural language understanding," in *2022 International Conference on Asian Language Processing (IALP)*, pp. 333–338, IEEE, 2022.

[3] S. Cahyawijaya, G. Winata, B. Wilie, K. Vincentio, X. Li, *et al.*, "Nusax: Multilingual parallel sentiment dataset for 10 indonesian local languages," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3492–3499, IEEE, 2023.

[4] M. R. Agam, "Klasifikasi berita bahasa indonesia dengan recurrent neural network (rnn)," in *Universitas Muhammadiyah Malang*, 2024.