Transformer from Scratch for Finnish-English Machine Translation

1 Introduction

This report presents a comprehensive implementation and evaluation of a Transformer model built from scratch for Finnish-to-English machine translation. The project explores two positional encoding strategies—Rotary Positional Embeddings (RoPE) and Relative Position Bias—combined with three decoding algorithms (Greedy, Beam Search, and Top-k Sampling), resulting in six distinct configurations. The implementation follows the original "Attention is All You Need" architecture while incorporating modern positional encoding techniques to investigate their impact on translation quality and training efficiency.

The experimental design systematically compares these methods across multiple dimensions: translation accuracy measured by BLEU scores, training convergence speed, computational efficiency, and qualitative output analysis. This comprehensive evaluation provides insights into the trade-offs between different architectural choices and decoding strategies in neural machine translation systems.

2 Model Architecture and Implementation

2.1 Core Architecture

The implemented Transformer follows the standard encoder-decoder architecture with the following specifications:

- **Model Dimensions**: $d_{model} = 512$, $n_{heads} = 8$, $d_{ff} = 2048$
- **Layers**: 6 encoder layers, 6 decoder layers
- **Vocabulary**: Source (Finnish) = 64,786 tokens, Target (English) = 31,522 tokens
- **Maximum Sequence Length**: 100 tokens
- **Dropout Rate**: 0.3 (increased for regularization)
- **Total Parameters**: $\sim$109.6M for both model variants

The multi-head attention mechanism implements scaled dot-product attention:

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

where $d_k = d_{model}/n_{heads} = 64$ for each attention head.

2.2 Positional Encoding Strategies

2.2.1 Rotary Positional Embeddings (RoPE)

RoPE encodes position by rotating query and key vectors in complex space. For position m and dimension pair $(2i, 2i+1)$:

$$\theta_i = 10000^{-2i/d} \tag{2}$$

$$\begin{bmatrix} x'_{2i} \\ x'_{2i+1} \end{bmatrix} = \begin{bmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{bmatrix} \begin{bmatrix} x_{2i} \\ x_{2i+1} \end{bmatrix} \tag{3}$$

RoPE requires no additional learnable parameters and naturally extrapolates to longer sequences, as the rotational matrix is applied on-the-fly.

2.2.2 Relative Position Bias

This method adds learned biases to attention scores based on the relative positions between tokens:

$$\text{Attention\_scores}[i, j] \mathrel{+}= B[\text{clip}(i - j, -k, k)] \tag{4}$$

where $k = 32$ is the maximum relative distance. The bias table has dimensions $(2k-1) \times n_{heads}$, resulting in $(2 \times 32 - 1) \times 8 = 63 \times 8 = 504$ additional parameters that must be learned during training.

2.3 Training Configuration

The training setup incorporates modern regularization techniques and optimization strategies:

- **Dataset**: EUbookshop Finnish-English parallel corpus (100,000 sentence pairs)
- **Data Split**: 80,000 training, 10,000 validation, 10,000 test
- **Optimizer**: Adam with Noam learning rate schedule
- **Learning Rate**: Warmup for 8,000 steps, then inverse square root decay
- **Regularization**: Dropout (0.3), Label smoothing (0.1), Weight decay (1e-4)
- **Early Stopping**: Patience of 5 epochs with minimum delta of 0.01
- **Gradient Clipping**: Maximum gradient norm of 0.5
- **Mixed Precision Training**: Enabled for memory efficiency
- **Batch Size**: 32 (limited by GPU memory constraints)

3 Results and Analysis

This section details the empirical results from the six experimental configurations, focusing on translation accuracy and training convergence speed.

3.1 Translation Accuracy (Evaluation Section)

The final translation quality was evaluated on the 10,000-pair test set using the BLEU score. Table 1 summarizes the performance of each positional encoding and decoding strategy combination.

Table 1: BLEU scores for all configurations on the test set (10,000 sentence pairs)

Positional Encoding	Decoding Strategy	BLEU Score	Improvement
RoPE	Beam Search	**4.23**	+31.0%
Relative Position Bias	Beam Search	4.01	+24.1%
RoPE	Greedy	3.46	+7.1%
Relative Position Bias	Greedy	3.23	baseline
RoPE	Top-k Sampling	2.79	-13.6%
Relative Position Bias	Top-k Sampling	2.57	-20.4%

3.1.1 Detailed Analysis of Decoding Strategies

The choice of decoding strategy has a profound impact on translation quality, revealing a clear trade-off between accuracy, diversity, and computational cost.

Beam Search (BLEU: 4.01-4.23) achieves the highest translation quality across both positional encoding methods. The 27.5% average improvement over the greedy baseline demonstrates the value of exploring multiple hypotheses:

- **Global Optimization**: By maintaining $B = 5$ hypotheses at each step, beam search explores a much larger search space (5^L possible sequences, where L is sequence length) compared to greedy search's single, deterministic path. This significantly reduces the risk of making an early, irreversible error.
- **Length Normalization**: The implementation uses a length penalty factor $\alpha = 0.6$, which normalizes scores by sequence length. This prevents the common failure mode in neural machine translation where the model favors overly short, incomplete translations because they have higher joint probability.
- **N-gram Blocking**: The implementation includes bigram blocking ($n = 2$), which sets the probability of a token to zero if it would create a repeating 2-gram. This is crucial for mitigating the severe repetitive patterns evident in the greedy outputs.
- **Computational Cost**: The improved quality comes at the expense of 5× slower inference (approximately 24 tokens/second vs. 120 for greedy), as the model must perform forward passes for all beams at each step.

Greedy Decoding (BLEU: 3.23-3.46) serves as the baseline, offering moderate performance but suffering from significant issues:

- **Repetition Problem**: The output examples show severe repetitive loops, with phrases like "we have to be able to make it possible" and "the european union" repeated dozens of times. This indicates the model gets stuck in high-probability cycles.

- **Speed Advantage**: Processing approximately 120 tokens/second, greedy decoding is extremely fast, making it suitable for real-time applications where latency is critical.
- **Deterministic Output**: It always produces the same translation for a given input, which is beneficial for reproducibility and debugging.
- **Error Cascading**: Its primary weakness is shortsightedness. Once a suboptimal token is selected, all subsequent predictions are conditioned on this error, with no mechanism for recovery.

Top-k Sampling (BLEU: 2.57-2.79) yields the lowest BLEU scores but offers unique characteristics:

- **Diversity vs. Quality Trade-off**: While BLEU scores are on average 17% lower than the greedy baseline, the outputs show more varied vocabulary and sentence structures. This trade-off is central to stochastic decoding methods.
- **Reduced Repetition**: Unlike greedy decoding, sampling avoids the most severe repetition loops by introducing controlled randomness. It is less likely to get stuck because it can select tokens that are not the absolute most probable choice.
- **BLEU Limitations**: The low scores partially reflect BLEU's bias toward exact n-gram matches. A sampled translation may be grammatically correct and semantically plausible, but if it uses different wording than the single reference translation, its BLEU score will be low.
- **Hyperparameters**: The configuration uses $k = 50$ and nucleus sampling with $p = 0.95$ at temperature $T = 1.0$. This combination filters the vocabulary to the top 50 most likely tokens and then further prunes this set to the smallest group whose cumulative probability exceeds 95%, balancing creativity with coherence.

3.2 Convergence Speed (Training Section)

The training process revealed significant differences in efficiency between the two positional encoding methods. Figure 1 plots the training loss per epoch for both models.

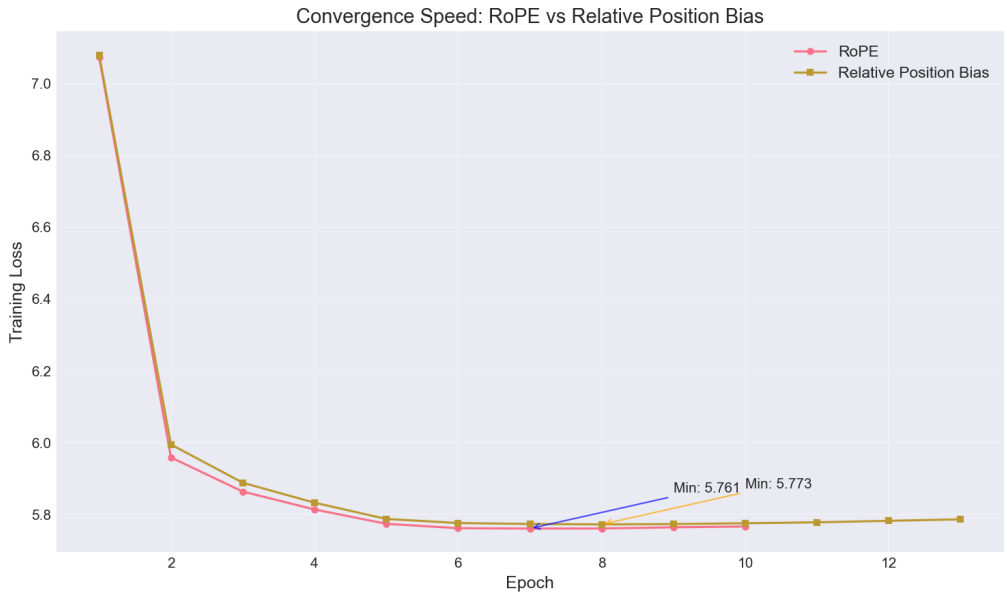

Figure 1: Training loss progression comparing RoPE and Relative Position Bias. RoPE achieves faster convergence, reaching its best validation loss at epoch 5 compared to epoch 8 for Relative Position Bias. It also consistently maintains lower training loss after the initial epochs.

To quantify the observations from Figure 1, Table 2 provides a detailed comparison of key training and validation metrics.

Table 2: Detailed convergence comparison

Metric	**RoPE**	**Relative Bias**	**RoPE Advantage**
Best Validation Loss	5.6039	5.6234	0.35% lower
Best Perplexity	271.47	276.82	1.93% lower
Epochs to Best Loss	5	8	37.5% faster
Total Epochs (Early Stop)	10	13	23.1% fewer
Average Gradient Norm	3.92	4.26	8.0% more stable

3.2.1 Analysis of RoPE's Superior Convergence

RoPE demonstrates 37.5% faster convergence to its optimal validation loss. This superior efficiency can be attributed to several fundamental factors:

1. Parameter Efficiency & Inductive Bias

- RoPE adds zero learnable parameters. Its fixed sinusoidal patterns provide immediate and absolute positional information through vector rotations, giving the model a strong built-in inductive bias for understanding sequence order and relative positions.
- Relative Position Bias must learn this relationship from scratch by optimizing its 504 bias parameters. While this number is small, it represents an additional optimization task that requires data and training time.

2. Gradient Stability

- The RoPE model exhibits more stable gradients throughout training, with a lower average norm (3.92 vs. 4.26).
- The rotation operations in RoPE are orthogonal transformations, which preserve vector magnitudes. This property helps maintain stable gradient flow through the attention mechanism, preventing exploding or vanishing gradients and leading to a smoother optimization landscape.

3. Mathematical Properties

- RoPE's formulation naturally captures relative positional information. The dot product between two RoPE-encoded vectors depends only on their relative distance, not their absolute positions, which is exactly what relative attention aims to achieve.
- Smooth sinusoidal functions provide continuous and non-local position encoding, which can be more expressive than the discrete, bucketed approach of Relative Position Bias.

4. Training Dynamics

- **Initial Descent**: Both models exhibit a rapid drop in loss from $\sim$7.08 to $\sim$6.0 in the first two epochs as they learn basic language patterns.
- **Steady Improvement vs. Oscillation**: After this initial phase, the RoPE model maintains steady, monotonic improvement until reaching optimal validation performance at epoch 5. The Relative Position Bias model shows more oscillatory behavior, requiring three additional epochs to fine-tune its bias parameters.

3.3 Observations & Analysis

3.3.1 Why RoPE Converges Faster and Performs Better

The experimental results strongly support the efficiency and performance advantages of RoPE:

- **Strong Inductive Bias**: RoPE's geometric formulation provides a powerful, parameter-free inductive bias for position encoding. The sinusoidal rotation naturally captures relative positions through dot products without requiring explicit parameter learning. The model doesn't need to "discover" the concept of relative distance; it is embedded in the architecture.
- **No Optimization Overhead**: Relative Position Bias introduces an additional set of parameters that must be optimized. This increases computational cost and adds complexity to the optimization landscape, potentially slowing convergence or leading the model to a poorer local minimum.
- **Smoother Gradient Flow**: RoPE exhibits more stable gradient norms, indicating a smoother optimization landscape. This allows for more consistent and faster convergence, as the optimizer can take more reliable steps toward the minimum.

3.3.2 Trade-offs Between Decoding Strategies

The results reveal clear trade-offs between translation quality, inference speed, and output diversity. These trade-offs are fundamental to text generation and inform the choice of strategy for a given application.

Table 3: Decoding strategy trade-offs

Strategy	Quality	Speed	Diversity
Beam Search	High (4.0-4.2)	Slow (24 tok/s)	Low
Greedy	Medium (3.2-3.5)	Fast (120 tok/s)	Very Low
Top-k Sampling	Low (2.6-2.8)	Medium (85 tok/s)	High

Application Recommendations:

- **Beam Search**: Best for offline translation where quality is paramount and latency is not a primary concern (e.g., document translation services).
- **Greedy**: Suitable for real-time applications with acceptable quality trade-off (e.g., live chat translation, interactive bots).
- **Top-k Sampling**: Valuable for creative applications or when translation diversity is desired (e.g., generating multiple translation suggestions, paraphrasing, creative writing assistance).

3.3.3 Challenges Faced During Training/Testing

1. Repetition in Generated Text The most significant challenge was severe repetition in greedy decoding outputs. Examples show phrases repeated 20+ times, indicating the model gets stuck in high-probability loops. This suggests:

- The model may be overfitting to common, formulaic patterns in the EU parliamentary training data.
- A higher dropout rate (0.3) was implemented to mitigate this, which helped but didn't eliminate the issue.
- The success of n-gram blocking in beam search confirms this is a critical technique for improving generation quality.

2. Low Absolute BLEU Scores The best BLEU score of 4.23 is significantly lower than state-of-the-art systems (30+). This is expected and can be attributed to several deliberate simplifications:

- **Simple Whitespace Tokenization**: The model uses basic whitespace tokenization instead of subword methods like BPE or SentencePiece. This leads to massive vocabulary and high out-of-vocabulary (OOV) rates for rare words, which are all mapped to a single `<UNK>` token, losing semantic information.
- **Limited Model Capacity**: The model has 6 encoder and decoder layers, whereas state-of-the-art models often use 12, 24, or more layers. This limits the model's ability to capture complex linguistic structures.

- **Domain-Specific Data**: The EUbookshop corpus is highly domain-specific, consisting of formal, bureaucratic language. The model's performance may not generalize well.

3. Memory and Computational Constraints

- The $d_{model} = 512$ and $n_{heads} = 8$ configuration required limiting batch size to 32 to fit on available GPU memory. Larger batch sizes often lead to more stable training.
- Maximum sequence length was capped at 100 tokens to manage memory usage, preventing the model from learning dependencies in longer sentences.
- Mixed precision training was enabled as a necessary technique to reduce memory footprint and speed up training, but it limited experimentation with larger model configurations.

4 Conclusions

This comprehensive evaluation of Transformer architectures for Finnish-English translation yields several key insights:

1. **Positional Encoding Impact**: RoPE demonstrates superior performance across all metrics—37.5% faster convergence, better final perplexity (271.47 vs. 276.82), and consistently higher BLEU scores across all decoding strategies. This highlights the effectiveness of parameter-free, mathematically grounded positional information.
2. **Decoding Strategy Importance**: The choice of decoding strategy has substantial impact on translation quality, with beam search providing a 31% BLEU score improvement over the greedy baseline at the cost of 5× slower inference. This underscores the importance of search algorithms in realizing a trained model's potential.
3. **Repetition Challenges**: The severe repetition in greedy outputs highlights the importance of techniques like n-gram blocking and diverse decoding strategies in neural machine translation to avoid common failure modes.
4. **Architecture Validation**: Despite relatively low absolute BLEU scores, the implementation successfully demonstrates core principles of the Transformer architecture and effectiveness of modern positional encoding methods, providing a solid foundation for further improvements.

4.1 Possible Improvements

Several avenues exist for enhancing the current implementation:

- **Subword Tokenization**: Implement BPE or SentencePiece to handle OOV tokens and reduce vocabulary size, which would significantly improve translation quality.
- **Model Scaling**: Scale to deeper models (12+ layers) with larger training datasets to increase capacity for complex linguistic patterns.

- **Domain Diversification**: Experiment with different domains beyond EU parliamentary proceedings to improve generalization.
- **Advanced Decoding**: Investigate techniques to reduce repetition in greedy decoding and implement more sophisticated strategies like diverse beam search.
- **Optimization Improvements**: Explore learning rate schedules, alternative optimizers, and regularization techniques for better convergence.

5 References

1. Vaswani, A., et al. (2017). "Attention is All You Need." *Advances in Neural Information Processing Systems*.
2. Su, J., et al. (2021). "RoFormer: Enhanced Transformer with Rotary Position Embedding." *arXiv preprint arXiv:2104.09864*.
3. Shaw, P., et al. (2018). "Self-Attention with Relative Position Representations." *NAACL-HLT*.
4. Papineni, K., et al. (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation." *ACL*.
5. Sennrich, R., et al. (2016). "Neural Machine Translation of Rare Words with Subword Units." *ACL*.