Positional Encoding in Transformers

Why Do We Need Positional Encoding?

The Problem: Attention is Position-Agnostic

The attention mechanism computes:

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

Key issue: This operation is **permutation-invariant** - it treats input as an unordered set, not a sequence.

Example

Without positional encoding, these sentences would have **identical representations**:

1. "The cat chased the dog"
2. "The dog chased the cat"
3. "Chased the cat the dog"

The model would compute the same attention weights regardless of word order, because it only sees word embeddings, not positions.

Why Position Matters

In language (and many other domains):

- **Word order determines meaning**: "dog bites man" ≠ "man bites dog"
- **Syntax depends on position**: subject comes before verb in English
- **Context is sequential**: pronouns refer to previous nouns
- **Temporal relationships**: "before" vs "after" depend on order

What Positional Encoding Does

Adds position information to word embeddings:

$$\text{Input} = \text{WordEmbedding} + \text{PositionalEncoding}$$

This allows the model to:

- Distinguish between same words at different positions
- Learn position-dependent patterns
- Understand sequential structure

What Would Happen Without Positional Encoding?

1. Loss of Word Order Information

Example: "not good" vs "good not"

- Without position: Both processed identically
- Model can't distinguish negation from affirmation
- **Result:** Semantic confusion

2. No Sequential Dependencies

Example: "Alice met Bob. She was happy."

- Without position: Can't tell "She" refers to Alice (who came first)
- Pronoun resolution fails
- **Result:** Lost coreference understanding

3. Broken Syntax Understanding

Example: "The dog that I saw yesterday barked"

- Without position: Can't parse nested clauses
- No subject-verb agreement tracking
- **Result:** Grammatical errors

4. Symmetric Attention Only

- Attention weights would be based purely on content similarity
- Can't learn patterns like "attend to previous words" or "look ahead"
- **Result:** No directional or relative position awareness

5. Poor Long-Range Dependencies

Example: "The keys that were on the table are missing"

- Without position: Can't track which "are" agrees with "keys" vs "table"
- **Result:** Long-distance agreement errors

Concrete Example

Input: "John loves Mary"

With positional encoding:

```
"John" → [word_emb] + [pos_0]
"loves" → [word_emb] + [pos_1]
"Mary" → [word_emb] + [pos_2]
```

Model learns: position 0 = subject, position 1 = verb, position 2 = object

Without positional encoding:

```
"John" → [word_emb]
"loves" → [word_emb]
"Mary" → [word_emb]
```

Model sees: three words with no order → can't determine who loves whom

Summary

Aspect	With Positional Encoding	Without Positional Encoding
Word Order	Preserved	Lost
Syntax	Learnable	Broken
Sequences	Understood	Treated as sets
"not good" vs "good not"	Different meanings	Identical
Coreference	Works	Fails
Performance	High	Severely degraded

Bottom line: Without positional encoding, transformers become **bag-of-words models** that can't understand sequential structure - essentially useless for language tasks.