1. Before the Transformer: How did computers read sentences?

Imagine a computer wants to understand a sentence:

“The cat sat on the mat.”

There were two older methods:

A. RNN (Recurrent Neural Network)

Think of RNN like reading a book one word at a time, slowly.

It reads word 1 → then word 2 → then word 3 → etc.

Like this:

```
The → cat → sat → on → the → mat
```

Problems:

Can’t read words in parallel

Forget what happened earlier in the sentence

If word 1 is connected to word 10 → too far, it forgets

Very slow to train

B. CNN (Convolutional Neural Network)

CNNs look at **groups of words at once**, not one by one.

Example:

```
[The cat] [cat sat] [sat on] [on the] [the mat]
```

Better than RNN, but still:

Needs many layers to understand long sentences

Still slower than needed

Still limited

Summary so far:

Method	How it reads	Problems
RNN	one word at a time	slow, forgets far words
CNN	groups of words	needs many layers

Both methods were too slow and bad at long sentences.

C. Then comes ATTENTION (Simple Meaning)

Attention means:

“Look at the whole sentence at once and focus on the important words.”

Example sentence:

“The animal didn’t cross the road because it was tired.”

Question:
What does “it” refer to?

A normal model struggles.
Attention does this:

“it” → looks at → “animal”
Not → “road”

So attention means:

✔ Look anywhere
✔ Jump to important words instantly
✔ No need to go word-by-word
✔ Very fast on GPUs

THEN THE PAPER SAYS:

“Why don't we build a whole model using ONLY attention
and remove RNNs and CNNs completely?”

This new model is the **Transformer**.

2. Now the paper will make sense (Simple Explanation)

⭐ The Transformer has two parts:

1. Encoder

Reads the input sentence.

2. Decoder

Creates/output the translation (word by word).

Just like Google Translate.

What does the Encoder do?

Imagine you give it the English sentence:

“The cat sat on the mat.”

The encoder:

- reads the whole sentence at once
- looks at connections between ALL words
- understands meaning
- outputs “smart representations” of the sentence

What does the Decoder do?

When translating to French:

“Le chat s’est assis sur le tapis.”

The decoder:

- generates one French word at a time

- but uses **attention** to look back at the English sentence

- example: when generating “assis” → look at “sat”

What makes Transformers special?

1. They read all words at the same time (parallel)

= very fast to train

2. They understand long sentences easily

= because any word can look at any other word instantly

3. No RNN, no CNN

= simpler, faster, better

This one idea changed EVERYTHING

All modern AI systems (ChatGPT, BERT, Bard, Llama) use the Transformer.

---

3. Now let me explain the important parts of the paper in beginner English

---

A. Self-Attention (the main idea)

Self-attention means:

A word decides which other words are important.

Example:

Sentence:
“The dog that lived next door barked loudly.”

When understanding “barked,”
the model focuses on “dog,”
not “door.”

This focus = attention.

B. Multi-Head Attention (simple meaning)

Instead of having 1 way to look at words…

The transformer uses **8 ways at the same time**.

Each “head” looks for something different:

- Head 1 → who is subject of verb
- Head 2 → who owns what
- Head 3 → which words describe a noun
- Head 4 → time relations
- Head 5 → locations
- …

All heads together = powerful understanding.

C. Positional Encoding (simple meaning)

Attention reads all words at once…
but it does NOT know order.

So we add a signal that represents:

- This is word 1
- This is word 2
- This is word 3
- …

Like giving each word a tag:

```
Word:  “cat”
Position: 2
```

This helps the model understand grammar.

D. Feed Forward Network (simple meaning)

After attention mixes information,
a small neural network “cleans up” the representation.

Think: polish the understanding.

E. Masking (simple meaning)

In translation, the model generates next word one by one.

So when predicting the 3rd word,
it should not look at the 4th word.

Masking hides the future.

F. Training tricks

They used:

- Adam optimizer (just a way to adjust learning)
- Learning rate that goes up, then down
- Dropout (randomly turning off some neurons to prevent overfitting)
- Label smoothing (don’t be too confident)

These make training stable.

G. Results

The Transformer beat all previous models in:

- translation quality
- speed
- training time

This is why it became the standard.

FINAL SUPER-SIMPLE SUMMARY OF THE WHOLE PAPER

The Transformer is a new model that:

- Reads all words at once
- Lets each word focus on other important words
- Uses many “attention heads” to learn different relationships
- Faster and better than old models
- Achieves state-of-the-art translation

- No RNN, no CNN, only attention

This paper changed AI forever.

Understanding the Model of the Transformers

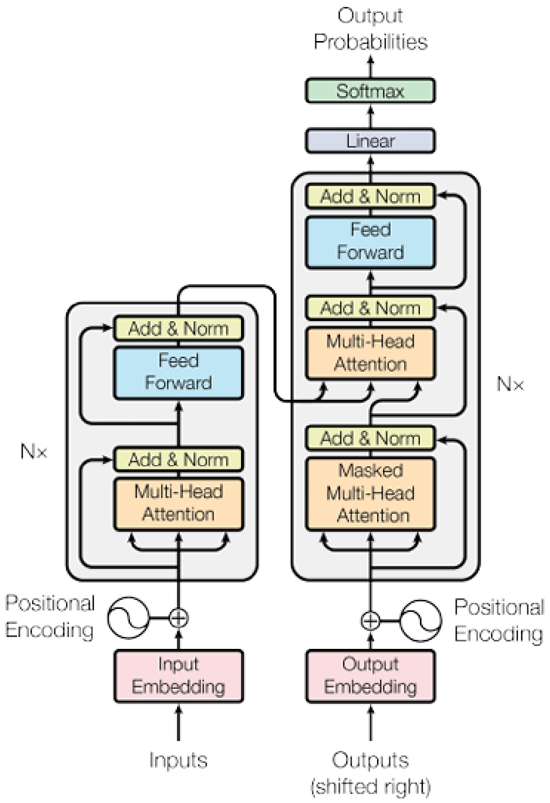

Figure 1: The Transformer - model architecture.

FIRST: What the diagram shows

This diagram is the **entire Transformer model** in one picture.

It has **two big sides**:

✔ Left side = Encoder (understands the input sentence)

✔ Right side = Decoder (creates the output sentence)

Think of it like:

- **Encoder = reading & understanding**
- **Decoder = writing/generating**

---

1 INPUT EMBEDDING (Bottom Left)

Pink box on left.

This is where the **input words** enter.

Example input:

```
"The cat sat"
```

Each word is turned into numbers (a vector).
 Words must be converted into numbers for the computer.

That pink box does that.

2 POSITIONAL ENCODING (Circle with + sign)

Under both Encoder & Decoder.

Attention does NOT understand word order.

So we add information about position:

- word 1
- word 2
- word 3

This gives the model a sense of order.

The **circle with +** means:

```
Word embedding + position info
```

---

3 ENCODER BLOCK (Left big stack repeated N times)

Labeled as “Nx”

“Nx” means this block is repeated **6 times** (in the original paper).

Each Encoder block has **2 layers**:

A. Multi-Head Attention (yellow box)

This is where the encoder looks at the whole sentence at once.

Example:

For “cat”, it checks:

- “the”
- “sat”

And learns their relationships.

B. Feed Forward (blue box)

A tiny neural network that improves the representation.

C. Add & Norm (grey box)

This is half-math, but simply:

- it stabilizes learning
- prevents errors
- keeps information flowing smoothly

Just think:

✔ “Add & Norm = tidy and clean the network.”

Arrows show connections within the block

Data flows:

```
Input → Multi-head attention → Add & Norm → Feed Forward → Add & Norm
```

After passing through 6 such blocks,
the encoder outputs a fully understood representation of the sentence.

---

4 DECODER BLOCK (Right big stack repeated N times)

Another “Nx” (also 6 layers

The decoder generates words **one by one**.

Each decoder block has **3 layers**:

A. Masked Multi-Head Attention (yellow-red box)

This is important.

Masked means:

Decoder **cannot look ahead**
(so it doesn't cheat and peek future words)

Example:
when generating word 3, it can only look at words 1 & 2.

This makes text generation work correctly.

B. Multi-Head Attention (yellow box)

Now the decoder looks at:

the encoder's output
+
previous decoder words

This is how it aligns words:

Example:

English input:

```
cat
```

French output:

```
chat
```

The decoder looks at "cat" (from encoder)
to generate "chat".

C. Feed Forward (blue box)

Same as in the encoder:
just a small network to refine understanding.

D. Add & Norm (grey boxes)

Again used to stabilize and clean information flow.

---

5 LINEAR + SOFTMAX (Top right)

Green and purple boxes.

After decoder finishes processing:

- The **Linear** layer converts numbers to vocabulary scores.
- The **Softmax** layer turns scores into probabilities.

This is where the model decides:

“Which word should I output next?”

For example:

```
chat = 0.92
chien = 0.01
maison = 0.007
...
```

The highest probability wins → output that word.

6.OUTPUT PROBABILITIES (Top)

Final output.

This is where the model outputs the translated word
(or the next word in any generation task).

SUPER SIMPLE SUMMARY OF THE WHOLE DIAGRAM

Below is the entire diagram explained like a story:

---

1. Encoder side

- Convert each input word to numbers
- Add position of word
- Pass through 6 layers:
    - attention checks relationships between words
    - feed-forward processes the meaning
    - norms keep it clean
- Output: fully understood version of input sentence

---

2. Decoder side

- Take previously generated words (shifted right)
- Mask future words so it cannot cheat
- Attend to itself
- Attend to encoder
- Process through 6 layers
- Convert to probabilities
- Output next word

Whole model image in three lines is:

Left side reads the sentence.
Right side writes the translation.
Attention connects everything together.