Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Attention is All You Need: Unpacking the Transformer Architecture that Rewired Artificial Intelligence

Chaitanya Dupad [1], Tarunkumar Dupad [2], Trupti Havaragi [3]

[1]*Student at Mountain House High School, Tracy*

[2]*Engineering Student at Aditya Engineering College, Bengaluru*

[2]*Engineering Student at Maratha mandal engineering college Belagavi*

ARTICLE INFO

Received: 12 May 2026

Revised: 02 July 2026

Accepted: 12 July 2026

ABSTRACT

Introduction: The publication of "Attention Is All You Need" by Vaswani et al. in 2017 marked one of the most consequential breakthroughs in the history of artificial intelligence. The paper introduced the Transformer architecture, discarding decades of reliance on recurrent neural networks (RNNs) and convolutional neural networks (CNNs) for sequence modeling in favor of a mechanism built entirely on self-attention. Where previous architectures processed tokens one step at a time, the Transformer allowed every token in a sequence to attend to every other token in a single parallel operation. This shift — computing $O(1)$ sequential operations instead of $O(n)$ — cut training time on machines with parallel hardware (GPUs/TPUs) from weeks to days, improved the modeling of long-range dependencies by shortening the maximum path length between any two tokens to a constant, and unlocked the scaling laws that now define modern AI development. The original base model trained on the WMT 2014 English-German translation task in just 12 hours on 8 NVIDIA P100 GPUs, reaching a new state-of-the-art BLEU score of 28.4 — a result earlier recurrent architectures required days to approach.The Transformer became the foundation of virtually every modern Large Language Model (LLM), including BERT, GPT-2/3/4, T5, PaLM, LLaMA, Claude, and Gemini, and extended well beyond text into Vision Transformers (ViT), Whisper (speech), AlphaFold 2 (protein structure), and multimodal systems that jointly reason over text, images, audio, and video.This article traces the evolution of sequence modeling that preceded the Transformer, walks through its internal mechanics with worked numerical examples, quantifies why attention is computationally and representationally superior to recurrence, and examines its lasting footprint on the AI industry.

Keywords: Attention Is All You Need, Transformer architecture, self-attention mechanism, Vaswani et al. 2017, what is a Transformer model, multi-head attention, positional encoding, encoder-decoder architecture, neural machine translation, deep learning architecture, Large Language Models, LLMs, Query Key Value attention, scaled dot-product attention, Recurrent Neural Networks, RNN, LSTM vs Transformer, vanishing gradient problem, BERT GPT T5 comparison, foundation models, generative AI architecture, natural language processing, NLP, sequence-to-sequence models, parallel processing in AI, scaling laws AI, GPT-4, GPT-3, BERT, T5, PaLM, LLaMA, Claude, Gemini, Vision Transformer, ViT, AlphaFold.

THE EVOLUTION OF SEQUENCE MODELING

1) Reading One Word at a Time (RNNs)

Older AI models called Recurrent Neural Networks (RNNs) worked like someone reading a sentence with a very short memory. They read one word, formed a quick impression, moved to the next word, updated that impression, and so on — always in order, never skipping ahead.

The problem: by the time the model reached the end of a long sentence, it had often "forgotten" details from the beginning. Imagine trying to remember the first sentence of a long email by the time you reach the last sentence —

Copyright © 2026 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

that's roughly the problem RNNs had. This is called the vanishing gradient problem: important signals from early words faded out the further the model read.

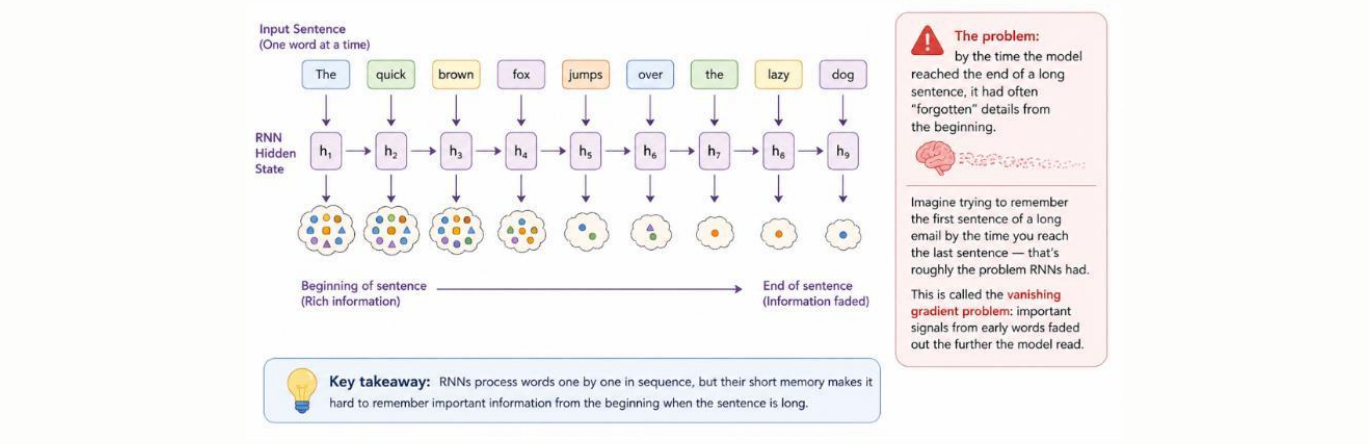

Figure 1: How RNN works – Reading one word at a time

2) A Better Memory (LSTMs)

In 1997, researchers introduced LSTMs (Long Short-Term Memory networks) — think of them as giving the model a notebook. As it reads, it can jot down important details, cross out things that no longer matter, and only look up entries when relevant. This "notebook" (called a cell state) had three simple controls:

- A forget gate — decide what to erase
- An input gate — decide what new info to write down
- An output gate — decide what to use right now

This helped a lot, and LSTMs became the standard for years. But there was still one big limitation: the model still had to read one word at a time, in order. You can't skip ahead. On a modern computer with thousands of processing cores ready to work in parallel, this is like insisting on using only one checkout lane at the grocery store even though 100 are open — slow by design.

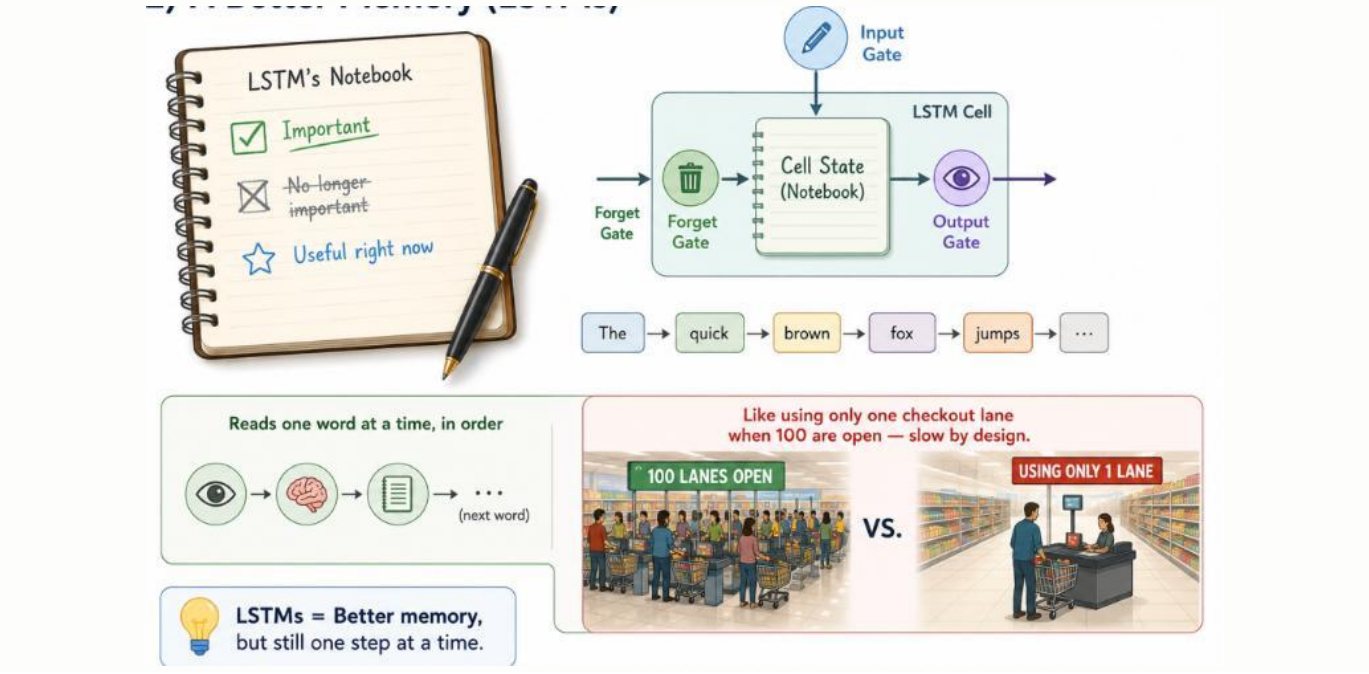

Figure 2: A Better Memory (LSTMs)

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

3) A Sneak Peek at Attention

In 2014, researchers (Bahdanau et al.) gave translation models a new trick: instead of trying to squeeze an entire sentence into one summary, let the model glance back at the original sentence while translating each word, and pay more attention to the most relevant parts. This was a huge improvement for translation quality — but it was bolted onto the old one-word-at-a-time RNN, so it was still slow.

The authors of the Transformer paper asked a bold question: if glancing back and forth is what actually helps, why keep the slow word-by-word reading at all? What if attention was the whole model?

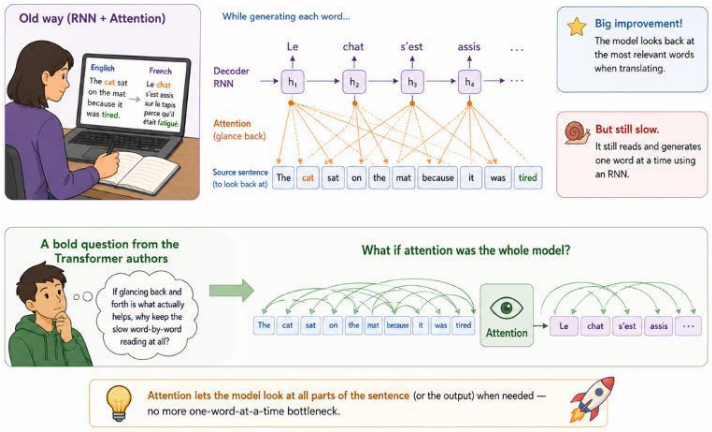

Figure 3: A Sneak Peak at Attention

1. The Transformer's Big Idea

The Transformer removes the "read one word at a time" step entirely. Instead, it looks at the entire sentence all at once, and for every word, it calculates how much attention to pay to every other word in the sentence — all in a single, parallel calculation.

A simple example: Take the sentence:

"The trophy didn't fit into the suitcase because it was too big."

What does "it" refer to — the trophy or the suitcase? A human instantly knows it's the trophy. An old RNN would have to carry that clue through 7 words of "memory," and it might get lost along the way. A Transformer instead directly compares "it" against every other word in the sentence at once, and learns that "it" relates most strongly to "trophy." One step, no memory decay.

Because this comparison happens all at once instead of step by step, the Transformer:

- Trains much faster — it can use all the parallel processing power of modern GPUs, instead of being stuck waiting for one word to finish before starting the next.
- Remembers long-range details better — connecting the first word and the last word of a paragraph takes exactly the same "distance" as connecting two neighboring words.
- Scales up dramatically — this is the property that let companies train models with billions, then trillions, of words of text.

To put a number on it: the original Transformer trained on a large translation dataset in about 12 hours using 8 GPUs, beating

older models that took days — while also producing more accurate translations.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

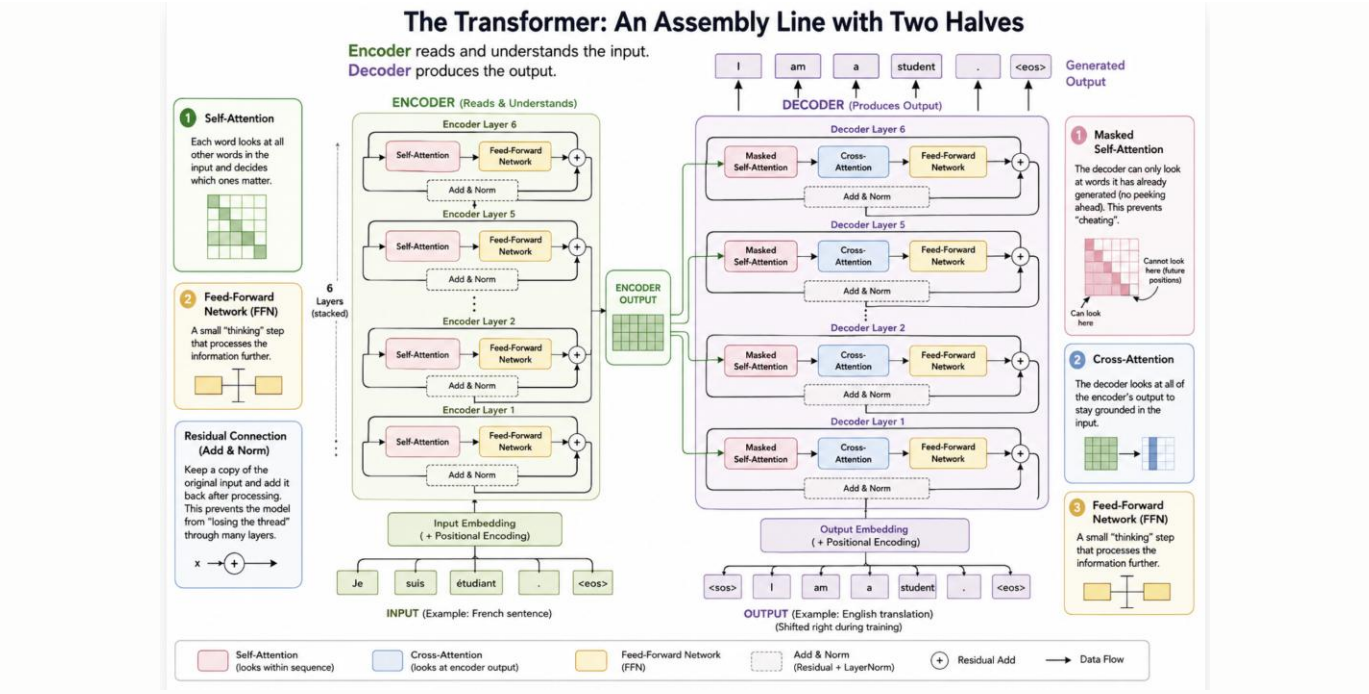

Figure 4: Transformer Architecture

HOW THE TRANSFORMER IS BUILT

Picture the Transformer as an assembly line with two halves: an encoder (reads and understands the input) and a decoder (produces the output, e.g., a translation or a chat reply). The original model stacked 6 encoder layers and 6 decoder layers on top of each other, like floors in a building — each floor refines the understanding a bit further.

Each encoder floor does two things to every word:

1. Self-attention — look at all the other words and decide which ones matter
2. A small "thinking" step (a feed-forward network) — process that information a bit further

There's also a simple but important trick called a residual connection: at each step, the model keeps a copy of the original input and adds it back after processing. This is like double-checking your notes against the original document — it prevents the model from "losing the thread" as information passes through many layers.

The decoder does the same two things, plus one more: it also looks back at the encoder's output, so it can stay grounded in what the input actually said (this is called cross-attention). It's also not allowed to peek at words it hasn't generated yet — otherwise it would be "cheating" by seeing the answer before writing it

SELF-ATTENTION EXPLAINED SIMPLY

Here's the core trick, in plain language. For every word, the model creates three simple representations:

- Query (Q) — "What am I looking for?"
- Key (K) — "What do I have to offer?"
- Value (V) — "Here's my actual content."

Think of it like a library search: your Query is your search term, every book has a Key (like a title or subject tag) that gets matched against your search, and the Value is the actual content inside the book you end up reading. Words whose "Key" matches the "Query" well get more attention; their "Value" contributes more to the final understanding of that word.

Mathematically, this is written as:

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

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

Don't worry about the formula — here's what it means in steps:

1. Compare the Query of one word against the Keys of every word (including itself) → gives a raw "relevance score" to every word.
2. Scale those scores down a bit (dividing by $\sqrt{d_k}$) so the numbers don't get too extreme — this just keeps the math stable during training.
3. Turn the scores into percentages that add up to 100% (this is what "softmax" does) — so now you have "attention weights," like: 38% attention on word A, 32% on word B, 30% on word C.
4. Blend the Values of every word together using those percentages as weights.

Worked mini-example: For the sentence "cat sat mat," when the model processes the word "sat," it might compute attention weights like:

cat: 38% sat: 32% mat: 30%

The final representation of "sat" becomes a blend: 38% of what "cat" contributes, 32% of what "sat" itself contributes, and 30% of what "mat" contributes. The model isn't just looking at "sat" in isolation anymore — it's looking at "sat" *in context*, informed by its neighbors, weighted by relevance. This whole calculation happens for every word, for the whole sentence, simultaneously — not one word after another.

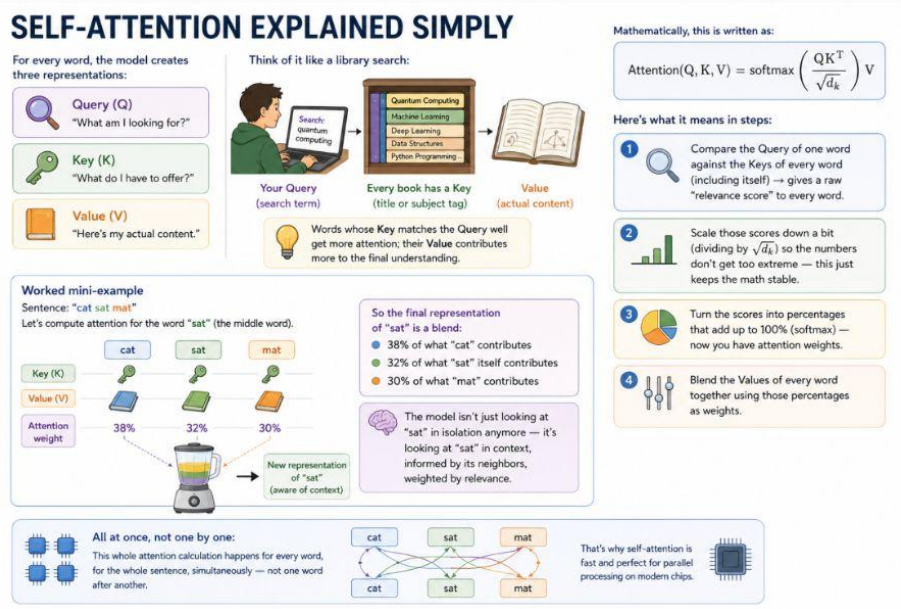

Figure 5: Self-Attention Diagram

MULTIPLE "PERSPECTIVES" AND WORD ORDER

1) Multi-Head Attention — Looking at the Sentence in Different Ways

Instead of doing this attention calculation just once, the Transformer does it 8 times in parallel, each time with a slightly different learned "lens." This is called multi-head attention.

Why do this? Because a sentence has many kinds of relationships at once. Imagine 8 different readers analyzing the same sentence, each with a different specialty:

- Reader 1 tracks grammar — which word is the subject of which verb.
- Reader 2 tracks pronouns — matching "it," "she," "they" back to the nouns they refer to.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

- Reader 3 tracks nearby words — simple local patterns.
- ...and so on.

Each "reader" (head) produces its own view, and at the end, all 8 views are combined into one richer understanding. This turns out to work far better than having just one generalist reader try to catch everything at once.

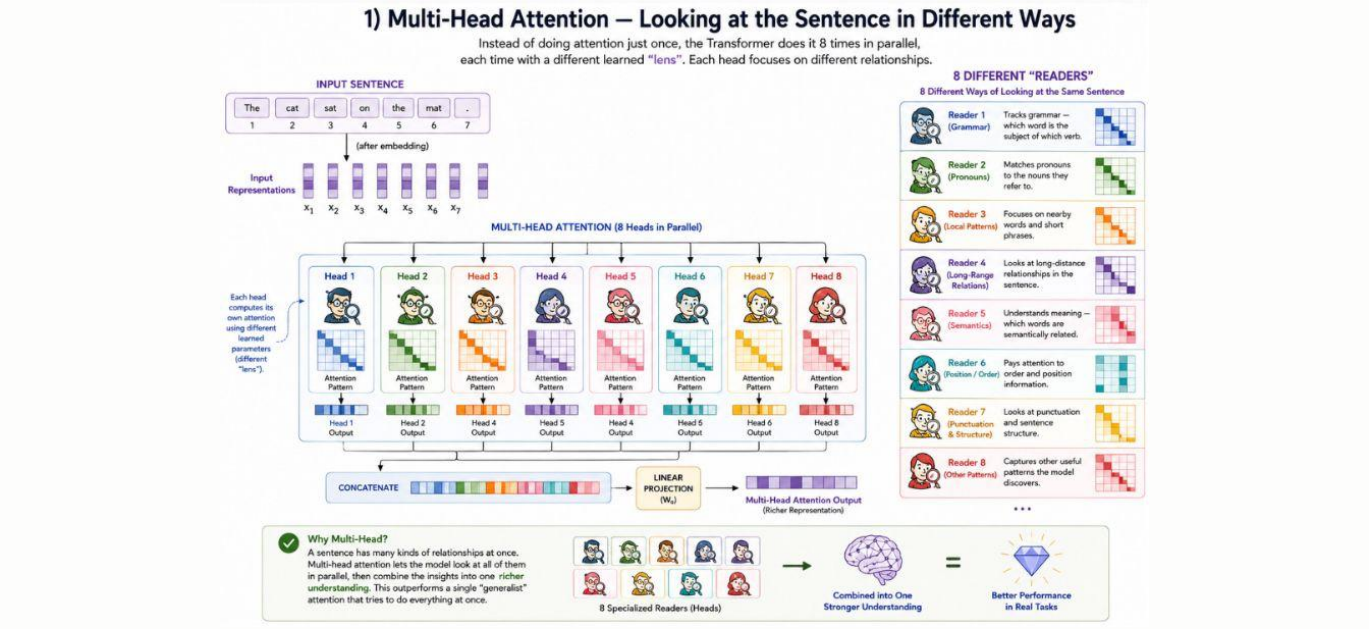

Figure 6: Multi-Head Attention — Looking at the Sentence in Different Ways

Positional Encoding — Remembering Word Order

There's one catch with looking at all words at once: the model has no built-in sense of order. "The dog bit the man" and "The man bit the dog" would look identical to pure attention, since it's just comparing words to each other regardless of position — a serious problem, since order changes meaning completely.

To fix this, the Transformer adds a small, unique numerical "stamp" to each word based on its position in the sentence (word 1, word 2, word 3, ...), using wave patterns (sine and cosine curves) at different frequencies. It's similar to how a barcode uses a unique pattern of lines to represent a number — each position gets its own recognizable pattern, which the model learns to interpret. This lets the model tell the difference between "dog bit man" and "man bit dog," even though it's technically comparing all words at once.

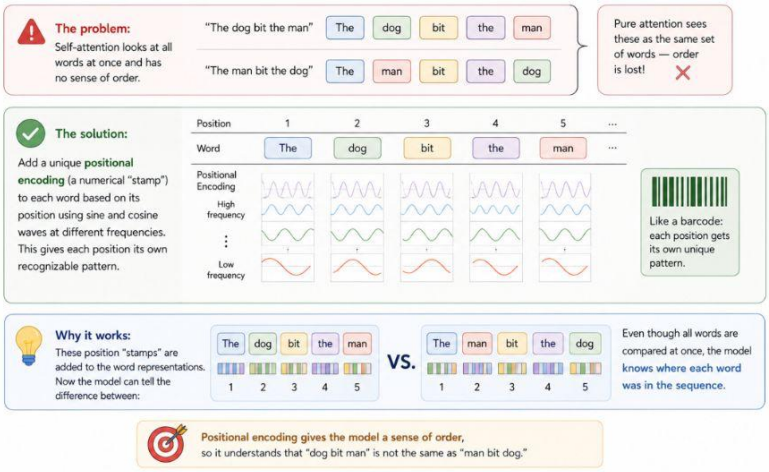

Figure 7: Positional Encoding - Remembering the word order

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

A FEW PRACTICAL TRAINING TRICKS

A few small, practical choices also mattered a lot for making the Transformer work well in practice:

- A gentle warm-up. The model starts training with a very low learning speed, gradually increases it, then slowly decreases it again. This is like easing a car into motion instead of flooring the gas pedal from a stop — it avoids the model "lurching" and destabilizing early in training.
- Not being too overconfident. During training, the model is deliberately discouraged from being 100% certain about any single answer (called *label smoothing*). This is a bit like teaching someone to say "I'm pretty sure it's X" instead of "It's definitely X" — it makes the model more reliable overall, even though it looks slightly less "confident" on paper.
- Breaking words into smaller pieces. Rather than needing a giant dictionary with every possible word, the model breaks less-common words into familiar sub-pieces (like breaking "unhappiness" into "un," "happi," "ness"). This lets it handle words it's never seen before.

These small engineering choices are now standard practice across nearly every large AI model trained today.

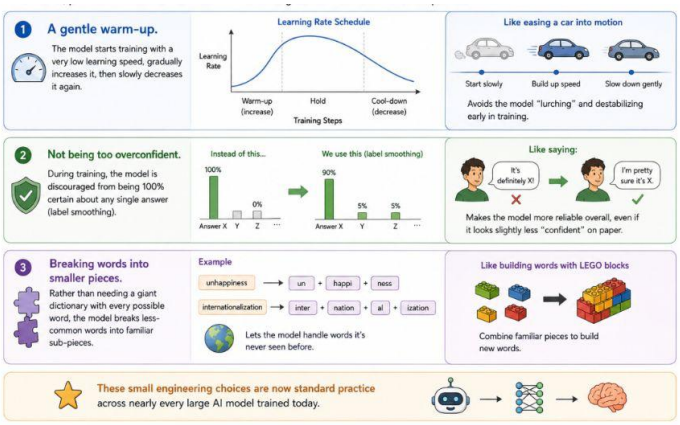

Figure 8: Training Example

WHY THIS CHANGED EVERYTHING

Because Transformers could be trained in parallel, researchers realized something important: the bigger you make these models, and the more text you train them on, the better and more predictably they perform. This discovery (called *scaling laws*) is the reason companies started racing to build bigger models — a race that simply wasn't possible with the old, slow, one-word-at-a-time architectures.

Here's how the idea spread across the AI world:

- BERT — used half of the Transformer (just the "reading" half) to deeply understand text for search engines and classification tasks.
- GPT models (like the ones behind ChatGPT) — used the other half (the "writing" half) to generate text one word at a time, but with each word informed by attention over everything written so far. GPT-3 scaled this up to 175 billion learned parameters.
- T5 — used the full encoder-decoder design to treat every language task (translation, summarization, Q&A) as one universal "text-in, text-out" format.
- Vision Transformer (ViT) — took the same idea and applied it to images, by chopping a picture into small square patches and treating each patch like a "word." It turns out attention works just as well on images as it does on text.
- Whisper, AlphaFold, and beyond — the same attention idea now powers speech recognition and even predicting how proteins fold in 3D — a problem that had stumped biologists for decades.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

- Claude, Gemini, LLaMA — today's leading AI assistants, which still use this same core attention mechanism, with refinements to make it faster and cheaper to run.

CONCLUSION

The real breakthrough of the Transformer wasn't a small accuracy bump — it was a fundamentally different way of processing sequences: look at everything at once and let the model learn what to pay attention to, instead of forcing it to read everything strictly in order. That one shift made models faster to train, better at handling long text, and — most importantly — able to scale up to sizes nobody thought possible in 2017.

Nearly every major AI system in use today, whether it's writing emails, generating images, transcribing speech, or folding proteins, still runs on this same core idea first published in "Attention Is All You Need."

GLOSSARY — ATTENTION IS ALL YOU NEED

A plain-language reference for the technical terms used in the article, listed alphabetically.

- Attention — A technique that lets a model look at all the words (or tokens) in an input and decide how much each one matters when trying to understand or generate a specific word. It answers the question: "Which other words should I focus on right now?"
- Attention Weight — A number (usually shown as a percentage) representing how much focus one word gives to another word. Weights across all compared words add up to 100%.
- BERT — A Transformer-based model (from Google, 2019) that only uses the "encoder" half of the architecture. Designed to deeply understand text, used heavily in search engines and text classification.
- BLEU Score — A metric used to measure how good a machine translation is, by comparing it to human-written reference translations. Higher is better.
- BPE (Byte-Pair Encoding) — A method for breaking words into smaller, reusable sub-word pieces (e.g., "unhappiness" → "un" + "happi" + "ness"), so the model can handle rare or unfamiliar words without needing a giant dictionary.
- Cross-Attention — A type of attention used in the decoder that looks back at the encoder's output, so the model stays grounded in the original input while generating a response or translation.
- Decoder — The half of the Transformer responsible for producing output (like a translation or a chat reply), one piece at a time, using both what it has generated so far and the encoder's understanding of the input.
- Encoder — The half of the Transformer responsible for reading and understanding the input text before any output is generated.
- Feed-Forward Network (FFN) — A small processing step applied to each word individually after attention, adding extra "thinking" capacity to refine the word's representation.
- Gradient — A signal used during training that tells the model how to adjust itself to reduce errors. Think of it as a correction hint sent backward through the network.
- Key (K) — In attention, a representation of a word that acts like a label or tag, used to determine how relevant that word is to a given search (Query).
- Label Smoothing — A training trick that discourages the model from being 100% overconfident about any single answer, which makes it more reliable in practice.
- Layer Normalization — A technique that keeps the numbers flowing through the network in a stable, consistent range, which helps the model train more smoothly.
- LLM (Large Language Model) — A very large AI model, built on the Transformer architecture, trained on huge amounts of text to understand and generate human language (e.g., GPT, Claude, Gemini).

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

- LSTM (Long Short-Term Memory) — An older type of neural network (1997) designed to remember important information over longer stretches of text using a "memory cell" with gates that control what to keep, forget, or use.
- Multi-Head Attention — Running the attention calculation multiple times in parallel (e.g., 8 times), each with a different learned focus, so the model can capture several kinds of relationships between words at once (grammar, pronouns, nearby words, etc.).
- Parallelization — Doing many calculations at the same time instead of one after another. The Transformer's biggest advantage is that it can process an entire sentence in parallel, unlike older models that had to go word by word.
- Positional Encoding — A mathematical "stamp" added to each word representing its position in the sentence (1st, 2nd, 3rd…), so the model can tell word order apart even though attention itself doesn't naturally track order.
- Query (Q) — In attention, a representation of a word that acts like a search request, used to find which other words (Keys) are most relevant to it.
- Recurrent Neural Network (RNN) — An older type of neural network that processes text one word at a time, in order, carrying forward a "memory" of what it has read so far. Slower and prone to forgetting earlier context.
- Residual Connection — A shortcut that adds a copy of a layer's original input back to its output, helping information and gradients flow through many layers without getting lost or degraded.
- Scaling Laws — The observed pattern that AI models get predictably better as they are made bigger and trained on more data and compute — a discovery that drove the race to build ever-larger models after the Transformer made this practical.
- Self-Attention — Attention applied within a single sequence, where each word compares itself to every other word in the same sentence (including itself) to build a context-aware understanding.
- Softmax — A mathematical function that converts a set of raw scores into percentages that add up to 100%, used to turn attention scores into attention weights.
- Token — A piece of text the model processes as one unit — this could be a whole word, part of a word, or a punctuation mark, depending on how the text was split up.
- Transformer — The neural network architecture introduced in "Attention Is All You Need" (2017) that relies entirely on attention (no recurrence or convolution) to process sequences, enabling parallel processing and better handling of long-range context.
- Value (V) — In attention, the actual content or information carried by a word, which gets blended together (weighted by attention scores) to form the final output.
- Vanishing Gradient Problem — An issue in older sequential models where training signals shrink to almost nothing as they travel backward through many steps, making it hard for the model to learn from information that occurred much earlier in a sequence.
- Vision Transformer (ViT) — An adaptation of the Transformer for images, where a picture is chopped into small patches that are treated like "words," allowing the same attention mechanism to process visual information.
- Warm-up (Learning Rate Warm-up) — A training technique where the model's learning speed starts low, gradually increases, then slowly decreases, to avoid instability early in training.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

REFERENCES

[1] Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS. https://arxiv.org/abs/1706.03762
[2] Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. Neural Computation.
[3] Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural Machine Translation by Jointly Learning to Align and Translate*.
[4] Devlin, J., et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers*.
[5] Brown, T. B., et al. (2020). *Language Models are Few-Shot Learners*.
[6] Kaplan, J., et al. (2020). *Scaling Laws for Neural Language Models*.
[7] Dosovitskiy, A., et al. (2021). *An Image is Worth 16×16 Words*.
[8] Raffel, C., et al. (2020). *Exploring the Limits of Transfer Learning with T5*.
[9] Chowdhery, A., et al. (2022). *PaLM: Scaling Language Modeling with Pathways*.
[10] OpenAI. (2023). *GPT-4 Technical Report*.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.