MECHANISMS OF ATTENTION ATTENTION & CONTEXTUAL EMBEDDINGS

Large Language Models are built on Transformers. A Transformer is a specific network architecture fundamentally based on the mechanism of Attention.

SEMINAL PAPER CITATION:

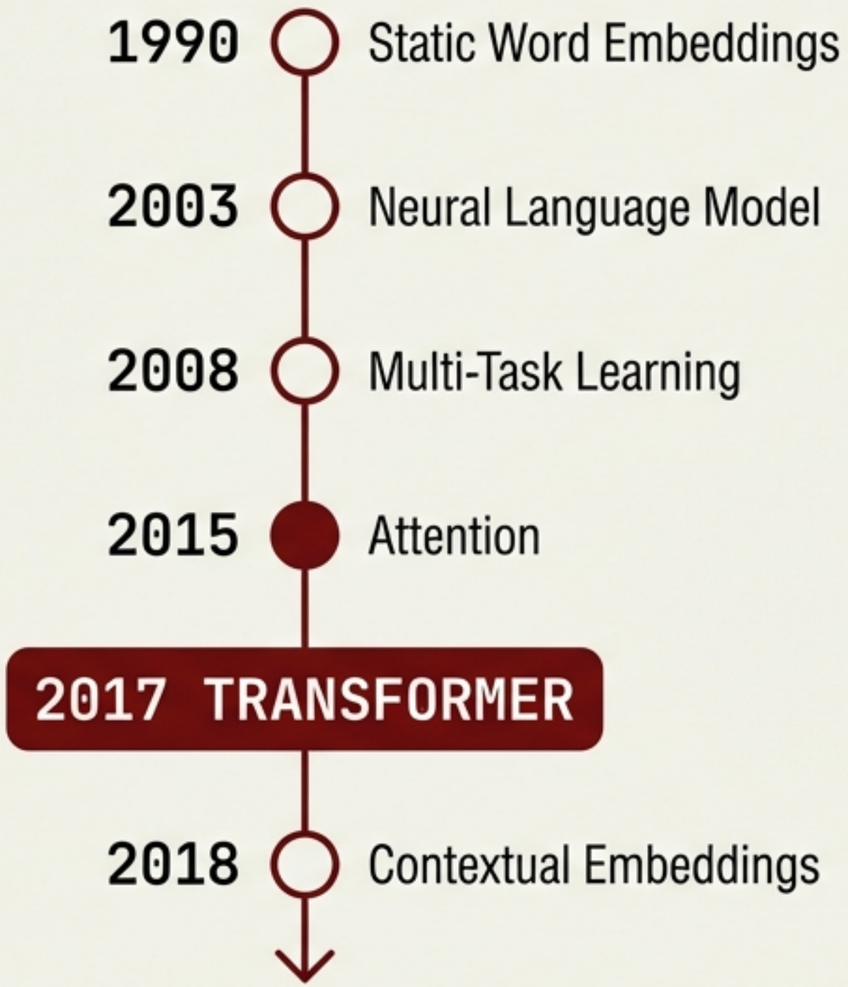
"Attention Is All You Need" (2017).

Ashish Vaswani Noam Shazeer Niki Parmar
Jakob Uszkoreit Llion Jones Aidan N. Gomez
Lukasz Kaiser Illia Polosukhin

Google Brain | Google Research | U of Toronto

NotebookLM

THE TIMELINE OF NLP ACCELERATION



For decades, language models relied on static definitions. The introduction of Attention Attention (**2015**) and **Transformers** (**2017**) **marked** the shift toward dynamic, context-aware understanding.

NotebookLM

THE PROBLEM: STATIC EMBEDDINGS

Systems like **word2vec** created static maps. The embedding for a word was fixed. It did not reflect how meaning shifts based on the sentence.

Dictionary (Fixed)
`"it": [0.2, -0.4, 0.9]`
(Immutable Vector)

THE AMBIGUITY:

The chicken didn’t cross the road because...

Does “it” mean chicken?
Does “it” mean road?

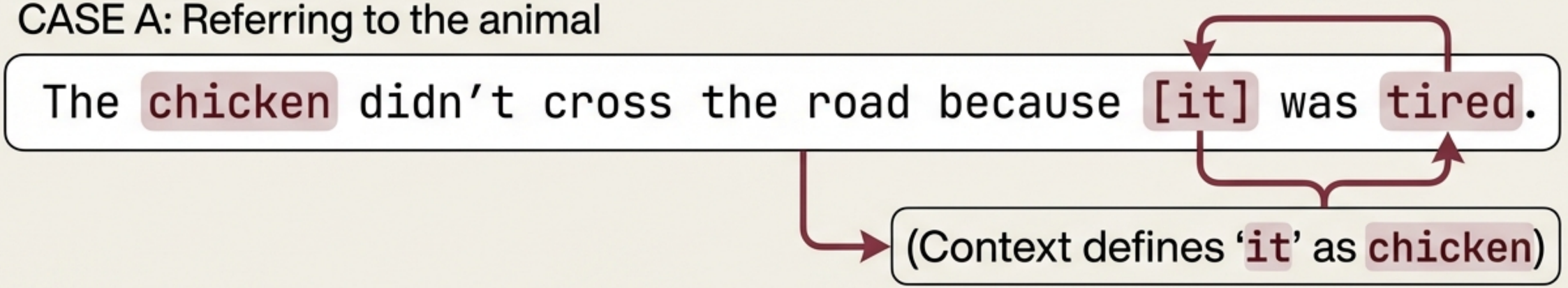
A static vector cannot know.

NotebookLM

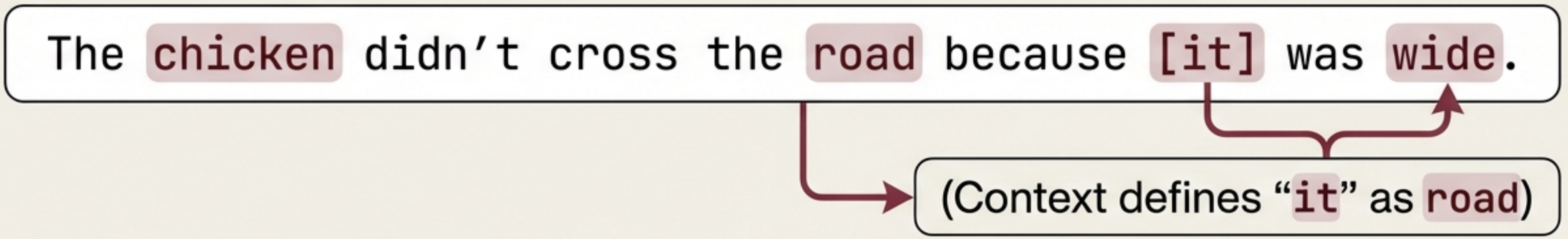
AMBIGUITY REQUIRES LOOKING AT NEIGHBORS

The identity of “it” is defined entirely by the context.

CASE A: Referring to the animal



CASE B: Referring to the street



NotebookLM

SOLUTION: CONTEXTUAL EMBEDDINGS

INTUITION

A word's meaning representation should differ across different contexts.

DEFINITION

Each word possesses a unique vector that expresses specific meaning derived from surrounding words.

How do we compute this?

↓

ATTENTION

The mechanism that allows a word to "look" at its neighbors before defining itself.

NotebookLM

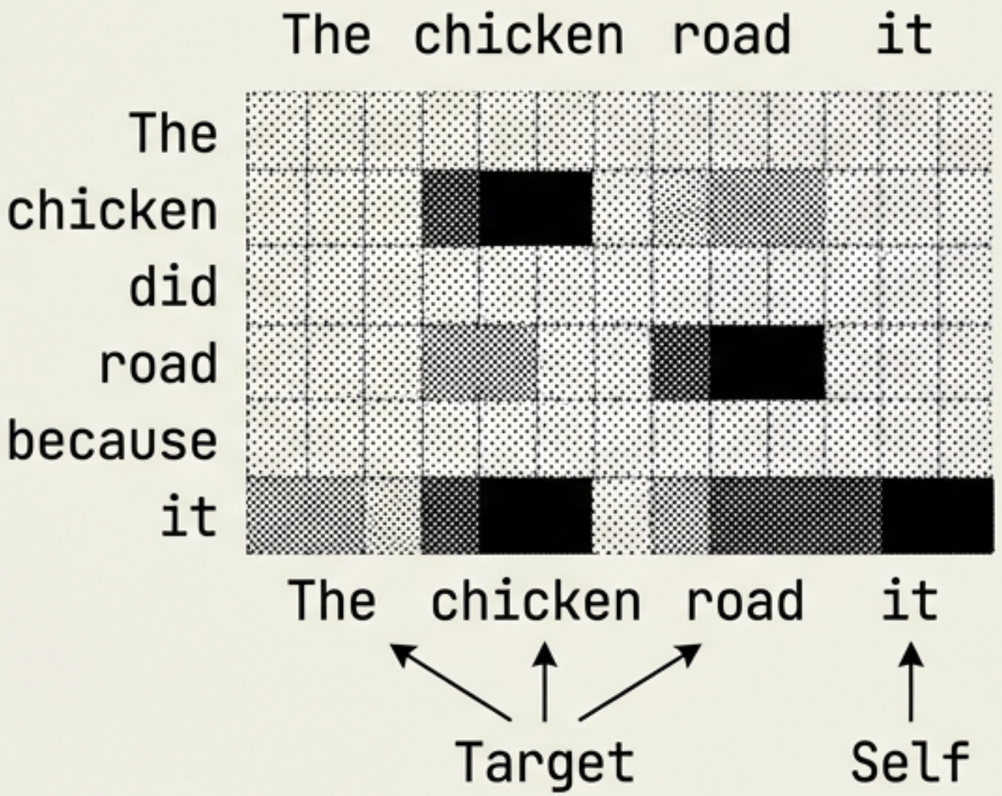
VISUALIZING ATTENTION

VISUALIZING ATTENTION

To build a contextual embedding, a word ‘attends to’ specific neighbors more than others. Selective integration of information.

Look at the column for ‘it’ (far right). It is attending strongly (dark blocks) to ‘chicken’ and ‘road’.

ATTENTION HEATMAP (Weights)



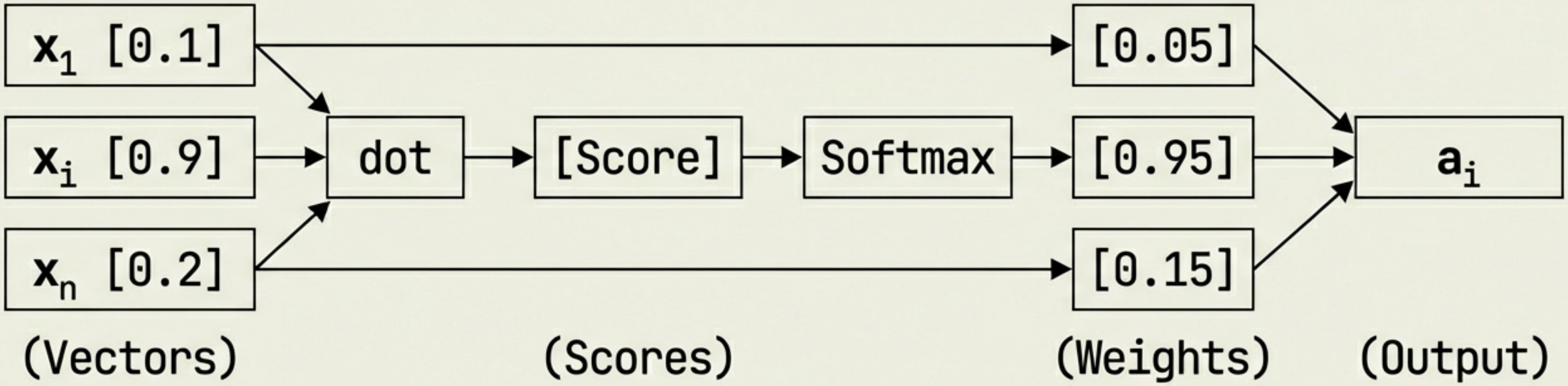
NotebookLM

MECHANISM: ATTENTION AS A WEIGHTED SUM

Goal: Produce output vector [a_i] from input vectors [x_j]

1. SIMILARITY → **2. NORMALIZE** → **3. SUM**

$\text{score} = \mathbf{x}_i \cdot \mathbf{x}_j$ $\alpha_{ij} = \text{softmax}(\text{score})$ $\mathbf{a}_i = \sum \alpha \cdot \mathbf{x}$



NotebookLM

THE THREE ROLES OF A TOKEN

Raw vectors are limited. In a Transformer, we project vectors into three distinct roles, like a database lookup.

🔍 QUERY (Q)

"The Agent"

What I am currently looking for.

e.g. "I am 'it', who defines me?"

🏷️ KEY (K)

"The Label"

How I identify myself to others.

e.g. "I am 'chicken', a noun."

📦 VALUE (V)

"Content"

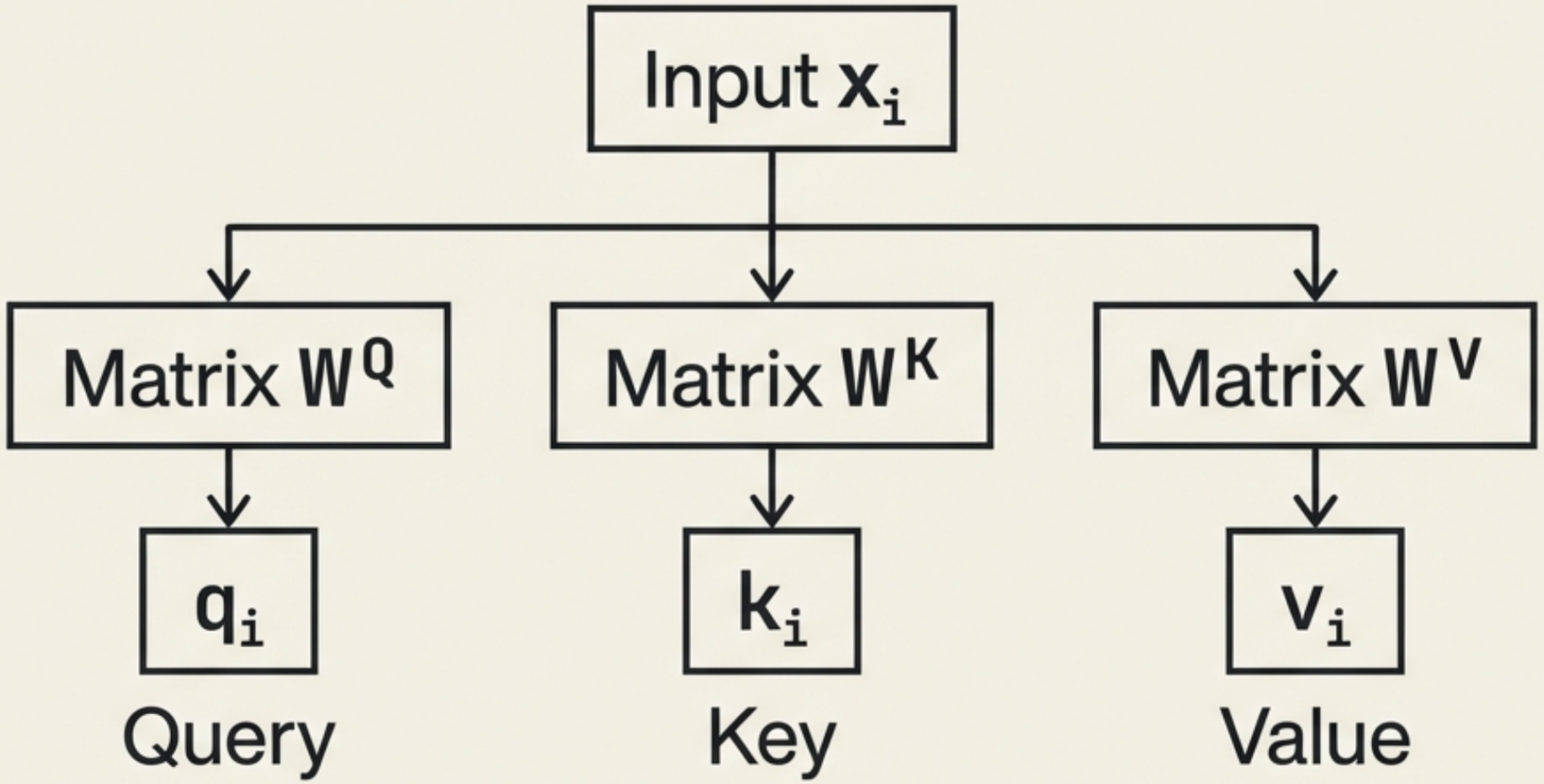
What I pass on.

The info payload.

NotebookLM

PROJECTIONS CREATE THE VECTORS

We use learned matrices to project input **x** into roles.



$$q = x \cdot W^Q \qquad k = x \cdot W^K \qquad v = x \cdot W^V$$

NotebookLM

THE MECHANISM OF ATTENTION

THE LOGIC

THE CORE EQUATION

1. COMPARE

Take the Query of the current token and Dot Product with Keys of context.

$$\text{score}(x_i, x_j) = \frac{q_i \cdot k_j}{\sqrt{d_k}}$$

2. NORMALIZE

Scale by root dimension size, then Apply Softmax to get probabilities.

$$\alpha_{ij} = \text{softmax}(\text{score})$$

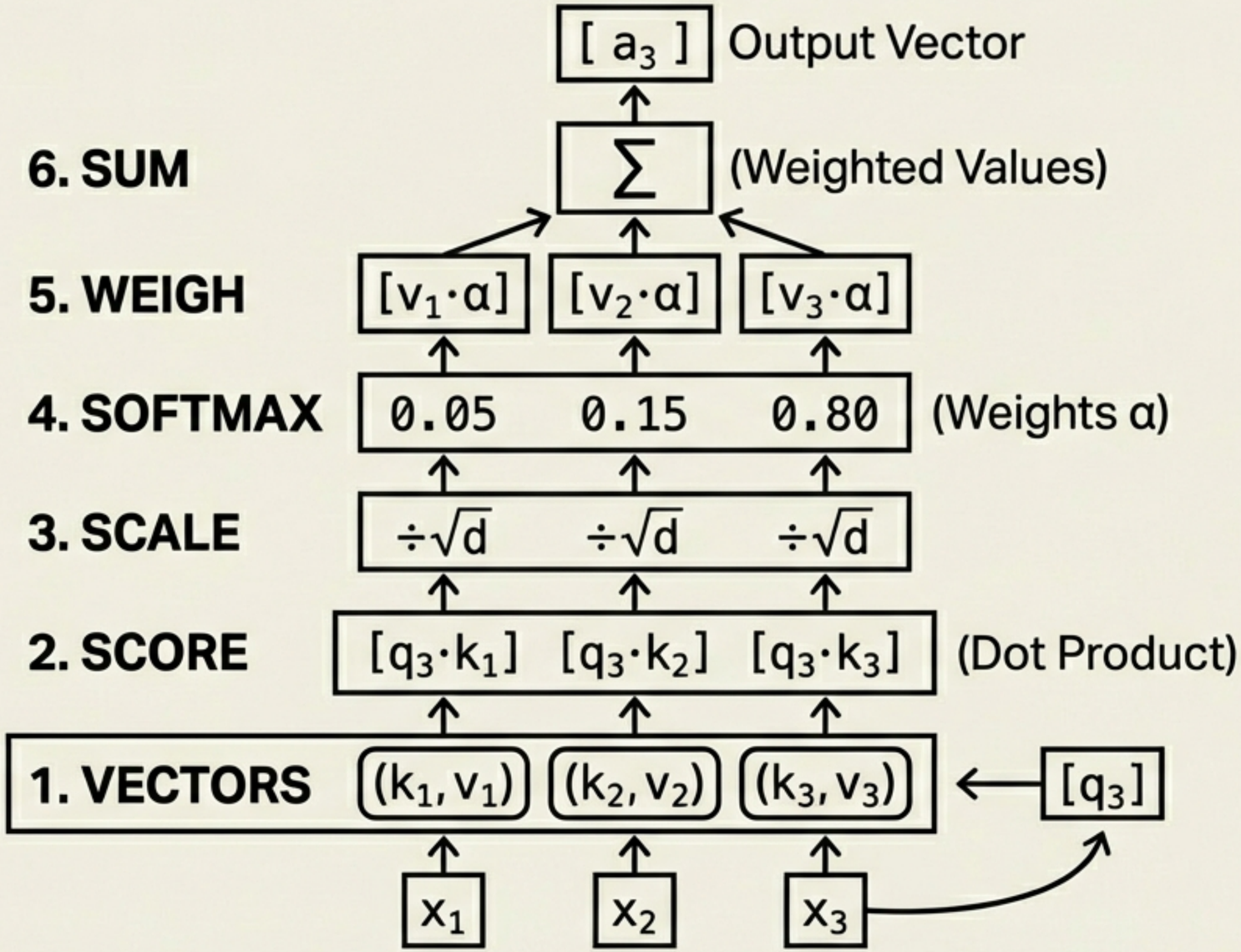
3. SUM VALUES

Multiply Value vectors by the weights and sum them.

$$a_i = \sum (\alpha_{ij} \cdot v_j)$$

NotebookLM

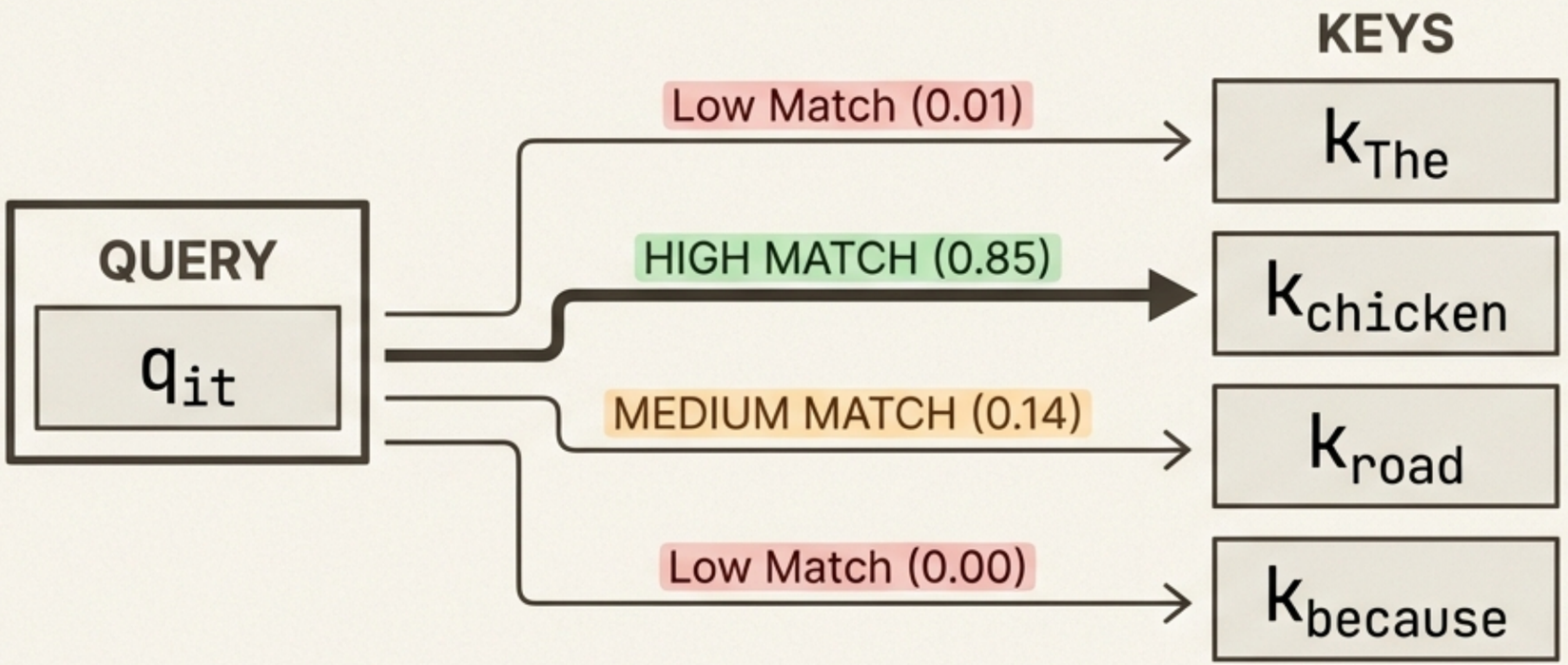
CALCULATING THE VALUE OF A TOKEN (a_3)



NotebookLM

VISUALIZING THE COMPARISON (Query vs Keys)

Comparing the Query for ‘it’ against all other Keys.



Result: The final definition of ‘it’ will be composed mostly of the Value vector from ‘chicken’.

NotebookLM

MULTI-HEAD ATTENTION

Concept & Equation

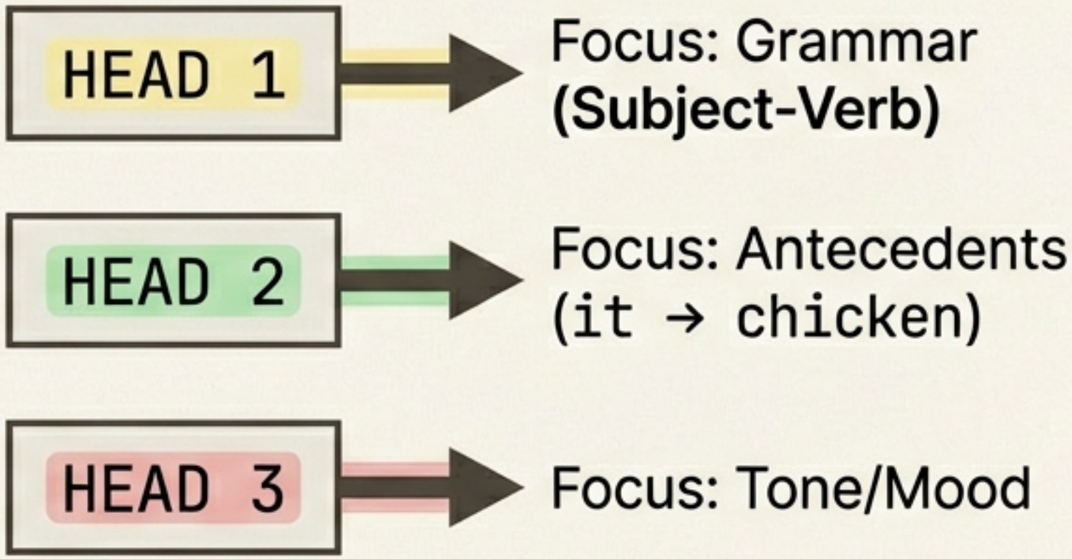
Transformers run multiple attention ‘heads’ in parallel.

$$\text{MultiHead}(x) = \text{Concat}(h_1 \ldots . h_h) \cdot W^O$$

Output Weight Matrix

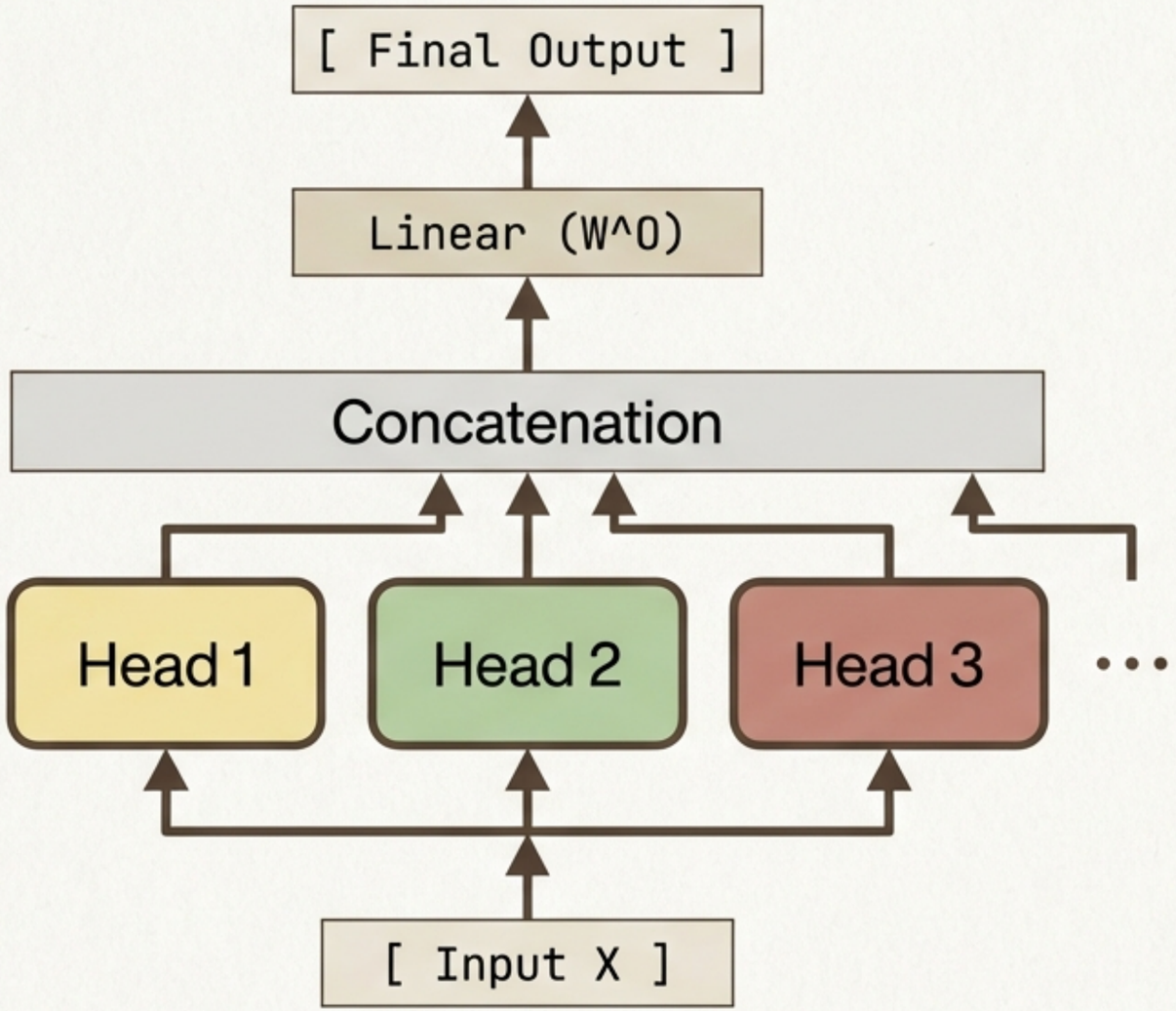
Intuition & Examples

Different heads look for different relationships.



NotebookLM

THE MULTI-HEAD ARCHITECTURE



NotebookLM

SUMMARY: RESOLVING CONTEXT

1. ENRICHMENT

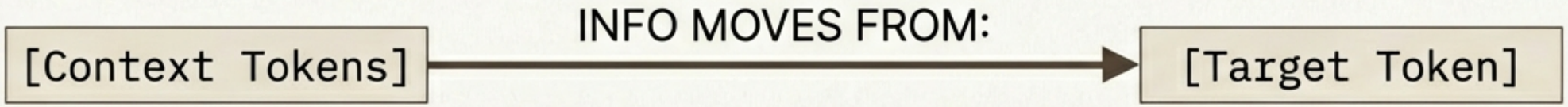
Attention enriches the representation of a token by incorporating information from its neighbors.

2. MECHANISM 2. MECHANISM

It projects words into Queries, Keys, and Values to calculate relevance scores.

RESULT 3. RESULT

The final embedding for “it” is no longer static—it is a composite of the word itself plus the relevant parts of “chicken” or “road.”



NotebookLM