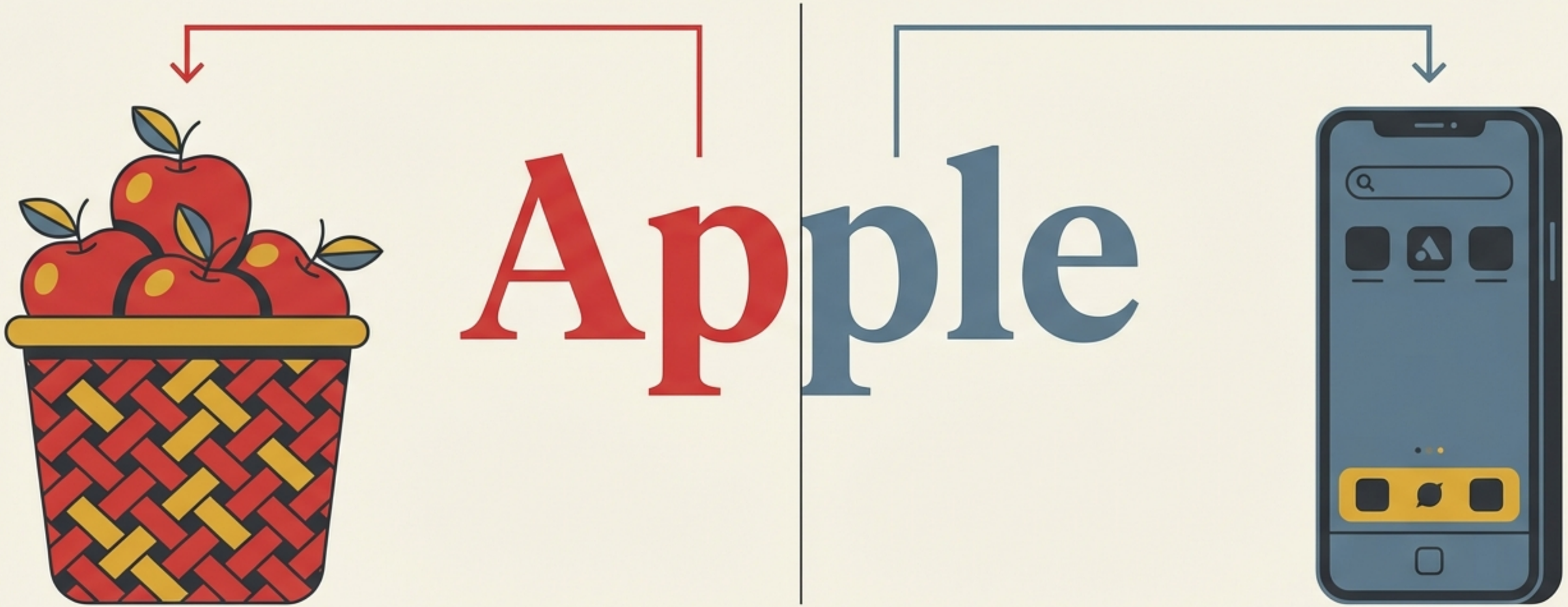
The Math Behind Attention

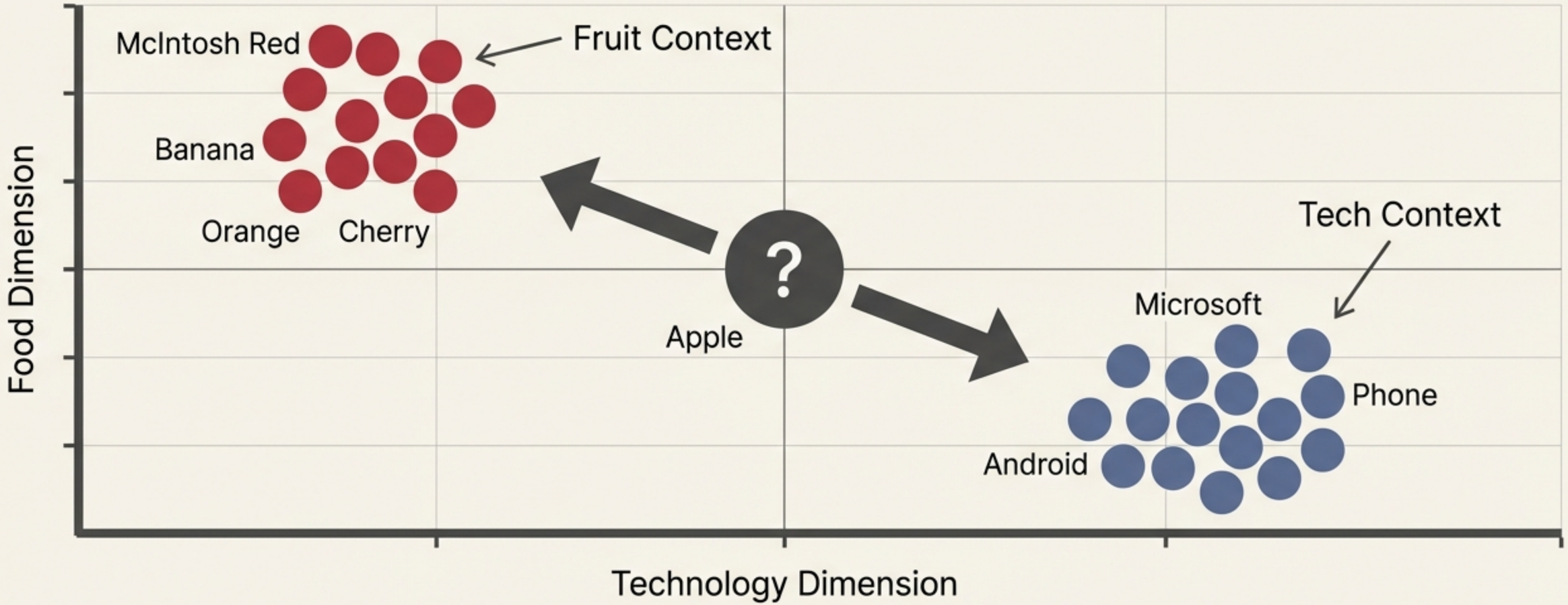
Deconstructing Queries (Q), Keys (K), and Values (V) using the “Apple” Paradox.



The Problem: Static Embeddings

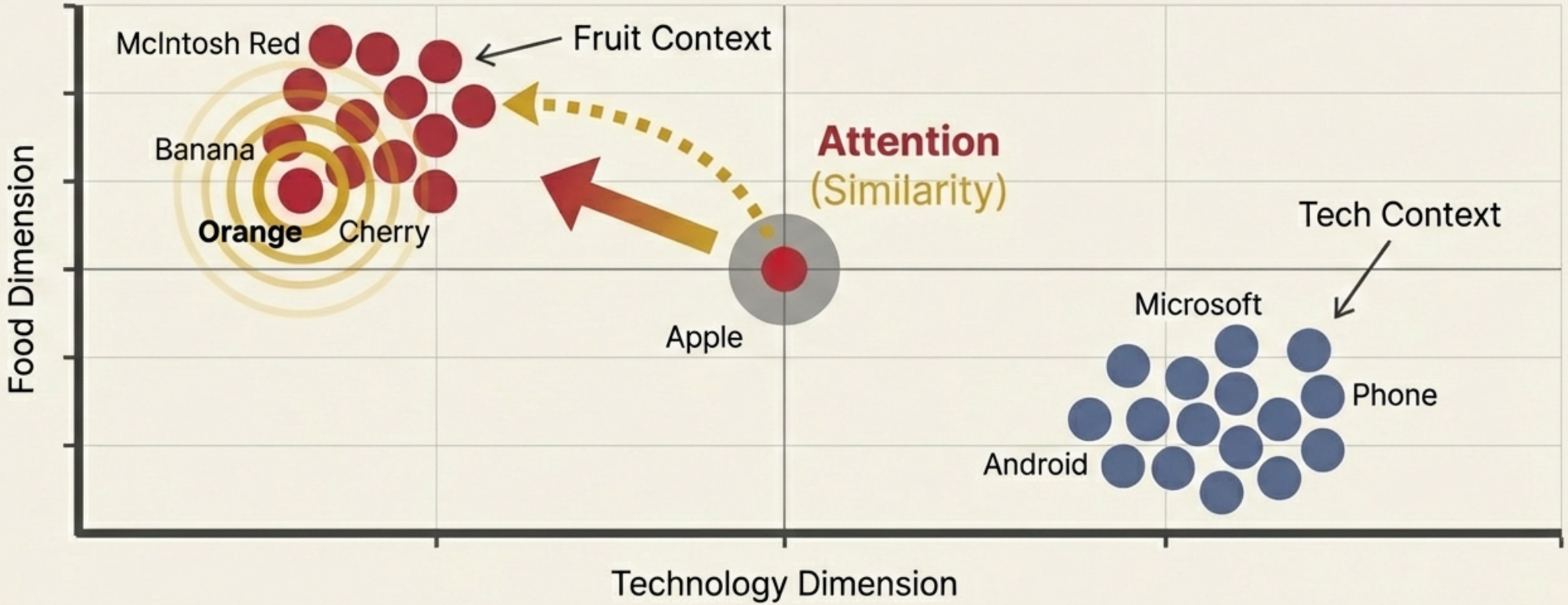
In traditional embeddings (like Word2Vec), a word has a fixed address. But "Apple" is homonymic—its meaning is fluid based on neighbors.

Scenario A: "I bought an apple and an **orange**." (Fruit)
Scenario B: "**Apple** unveiled a new **phone**." (Tech)



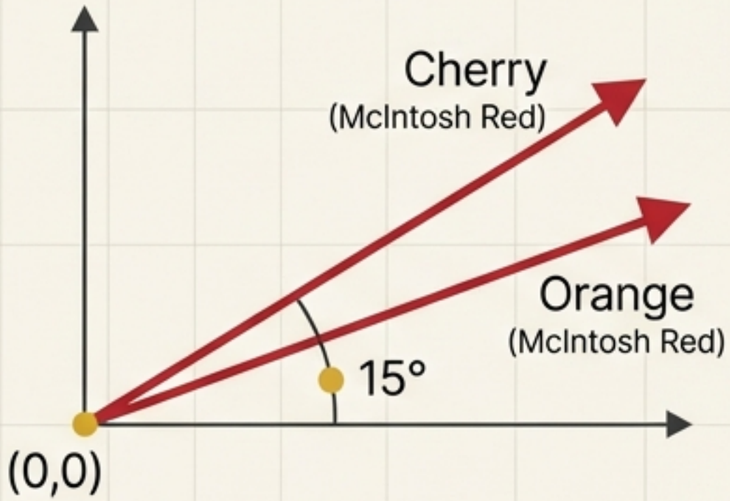
Attention is Contextual Gravity

In the sentence “an apple and an orange,” the word **Orange** acts like a magnet. It exerts a gravitational pull on **Apple**, moving its vector representation closer to the “Fruit” cluster.

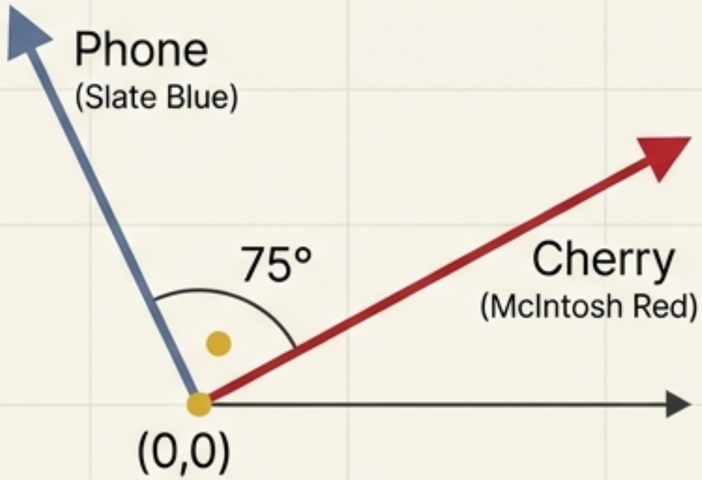


Measuring the Pull: The Dot Product

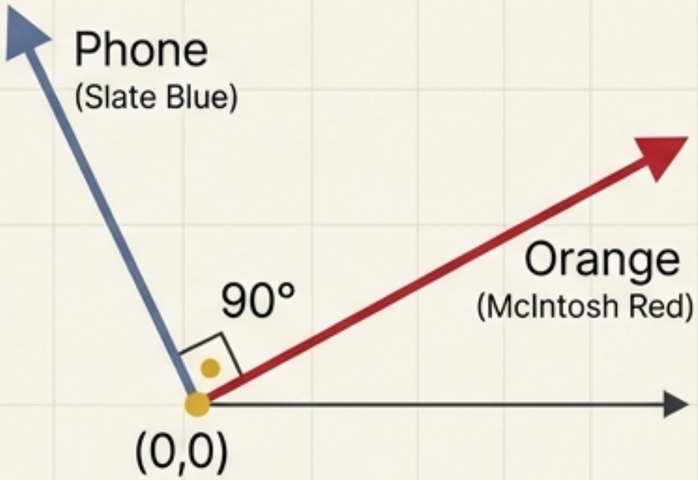
The model calculates alignment between vectors. Closely aligned vectors yield high attention scores. Orthogonal vectors yield zero attention.



High Dot Product
(High Similarity)



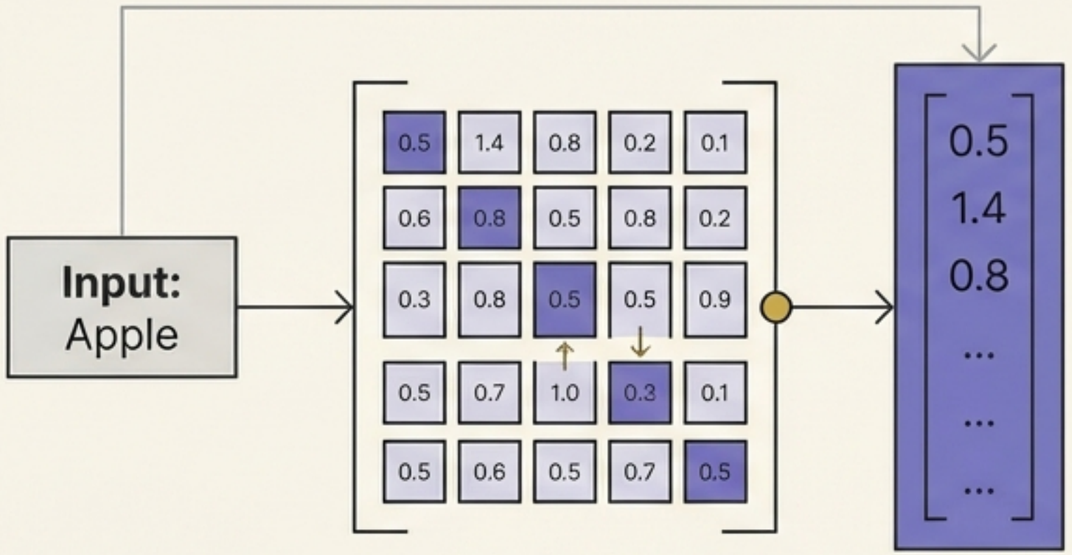
Low Dot Product
(Low Similarity)



Zero Dot Product
(Orthogonal / No Similarity)

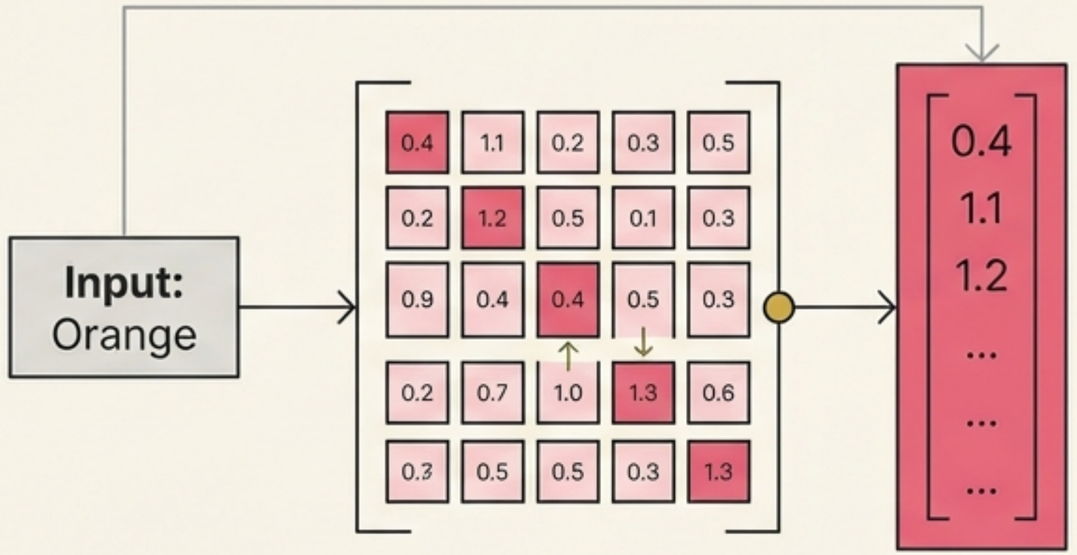
The Search Party: Queries (Q) and Keys (K)

To calculate attention, we project input words into a “Similarity Space.”
Q is the question the word asks. K is the label the word wears.



W_q
(Query Weights)

Query (Q):
What am I looking for? (Fruits? Tech?)



W_k
(Key Weights)

Key (K):
What defines me? (I am a fruit)

Calculating the Attention Score

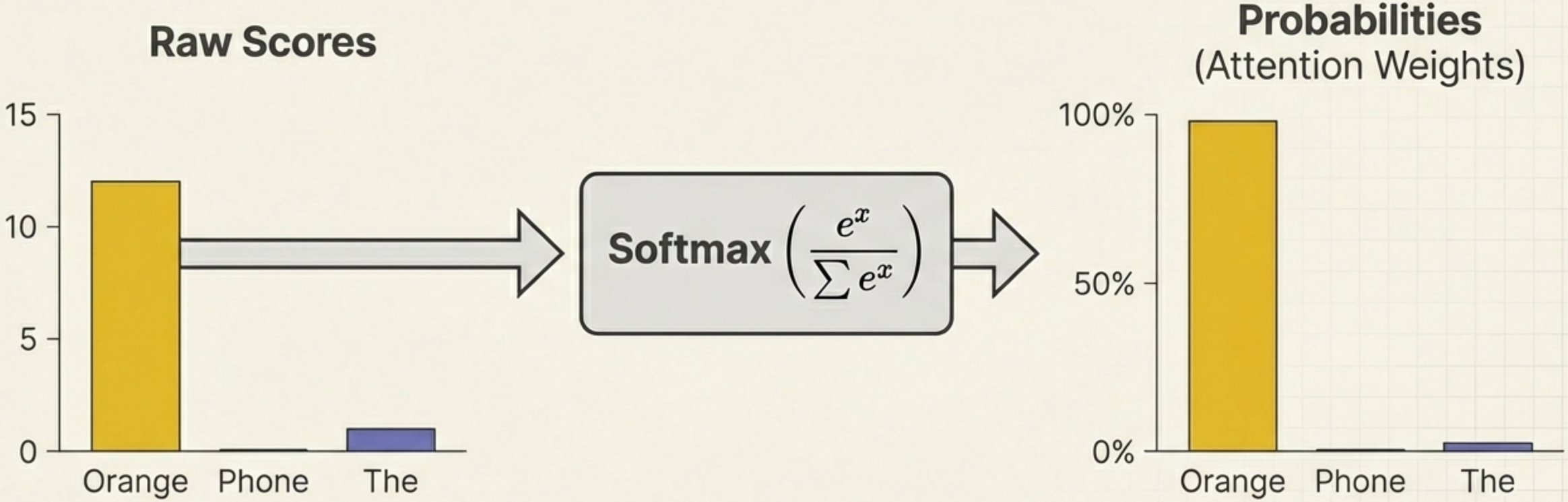
We calculate the dot product between the Query of “Apple” and the Keys of every other word. A high score means “Orange” holds the relevant context for “Apple”.

	Apple (K) Slate Blue	**Orange (K)** Slate Blue	**Phone (K)** Slate Blue	**The (K)** Slate Blue
Apple (Q) McIntosh Red	1.5	**12.0**	0.0	0.3

$$Score = Q \cdot K^T$$

Softmax: From Scores to Probabilities

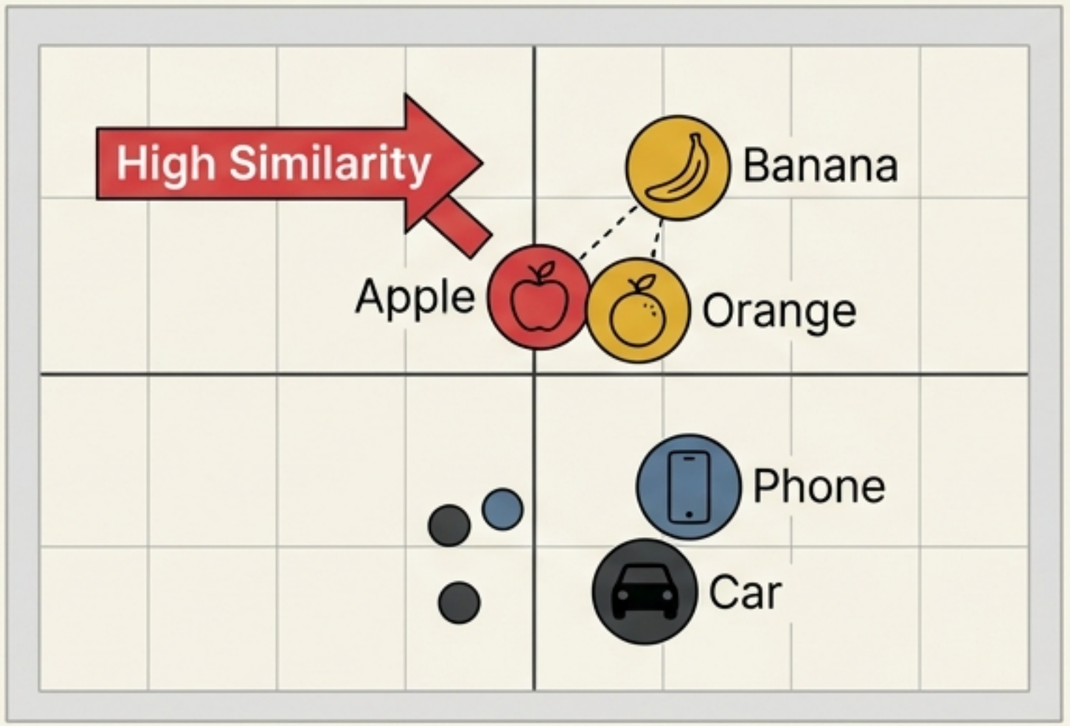
Raw scores are unruly. Softmax forces them into a distribution that sums to 100%. The model is told: "Take 98% of your information from Orange."



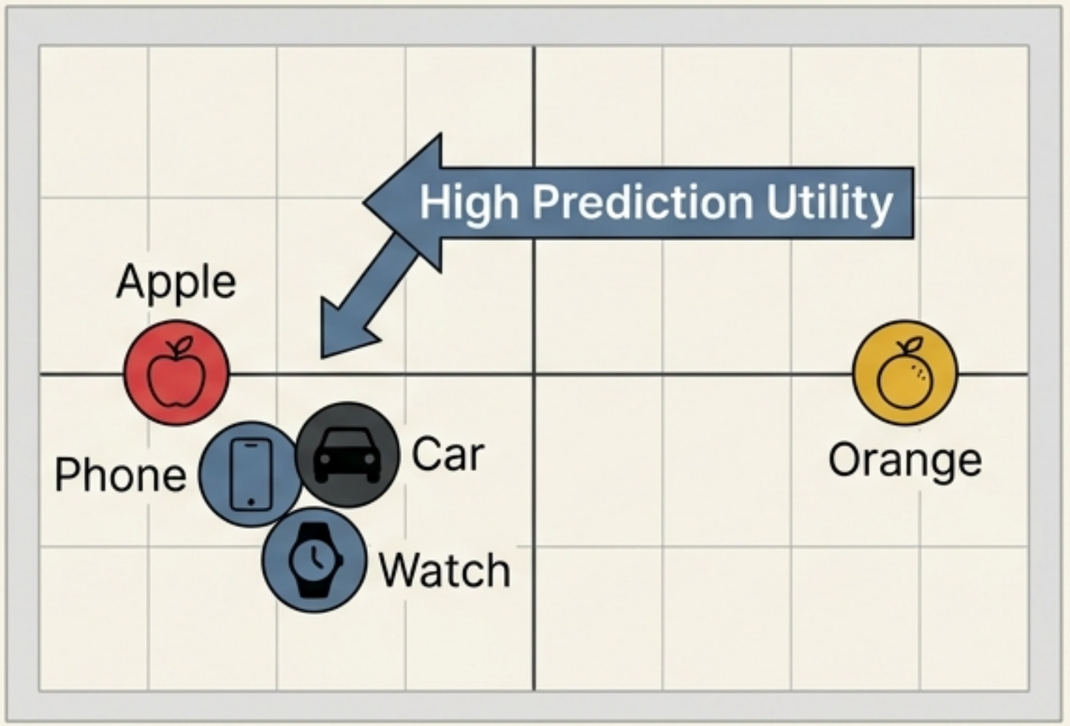
The Critical Distinction: Matching vs. Content

Why a separate Value (*V*) vector? The features that make words *similar* (Q/K) are often different from the information needed to *predict* what comes next (V).

Q/K Space: Optimized for Matching Features (Color, Shape, Category)

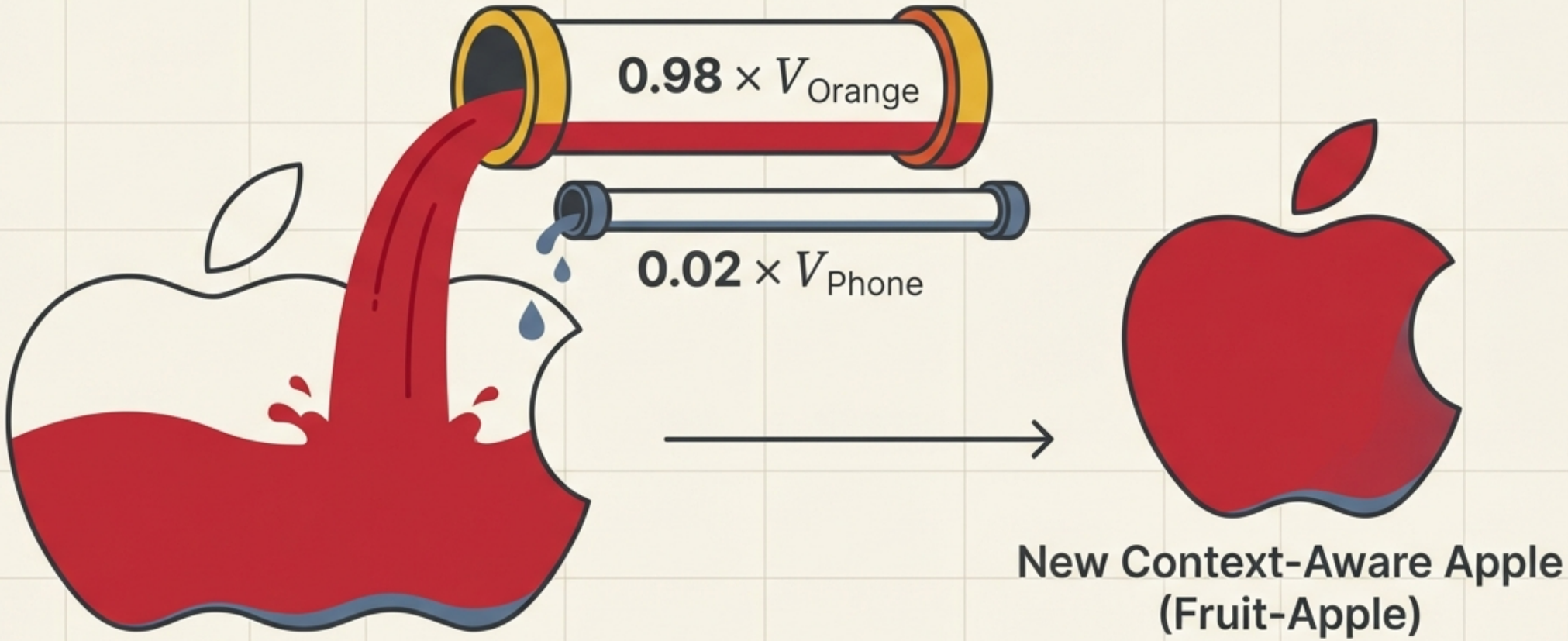


V Space: Optimized for Next Word Prediction (Grammar, Utility)



Values (V): The Information Carrier

Once we know “Orange” is the winner (98%), we don’t copy its Key. We extract its Value (V). We add 98% of the “Orange-Value” to the Apple representation.



Weighted Sum Update: $z = \sum \alpha_i V_i$

Summary: The Full Attention Loop

1. **Identify** relevant neighbors ($Q \cdot K$).
2. **Normalize** to probabilities (Softmax).
3. **Extract** useful content (V).
4. **Update** the word's meaning.