Self-Attention Mini-Exercise (Contextual Embeddings by Hand) How Transformers Turn Static Embeddings into Contextual Embeddings

How this handout connects to the slides

In the slides, we moved from **static embeddings** to **contextual embeddings**. A static embedding gives each word one fixed vector. A transformer improves on this with **self-attention**: each token compares itself with the other tokens, turns those comparisons into attention weights, and then forms a weighted sum to create an updated embedding that depends on context.
The goal of this in-class exercise: do one complete, simplified self-attention pass by hand for an ambiguous word. No calculus, no coding, and only light arithmetic.

Why this matters in business settings. The same term can mean very different things across customer reviews, search queries, support tickets, contracts, and earnings-call transcripts. Contextual embeddings help modern NLP systems decide which meaning is intended.

Toy setting (for easy arithmetic). We focus on the target word **bank** in two short contexts:

Context A: "river fishing bank" Context B: "money deposit bank"

We ignore small filler words and keep only **two semantic dimensions**:

Semantic dimension 1: Nature Semantic dimension 2: Finance

Each word is represented by a tiny embedding of the form [Nature, Finance].

Quick toolbox

Dot product reminder. $[a, b] \cdot [c, d] = ac + bd$.

Simplified self-attention recipe for this exercise.

1. **Similarity:** compute a relevance score using a dot product, $x_{\text{bank}} \cdot x_i$.
2. **Scaling:** divide each similarity score by 2 to keep the values in a reasonable range.
3. **Softmax:** turn the scaled scores into **attention weights** that add up to 1.
4. **Weighted sum:** combine the token embeddings using those weights:

$$y_{\text{bank}} = \sum_i w_i x_i$$

Business translation. Similarity = relevance score; Scaling = keep scores from becoming too extreme; Softmax = attention shares; Weighted sum = updated contextual embedding.

Note. In a full transformer, scaling uses $\sqrt{d_{\text{model}}}$. Here we divide by 2 to keep the arithmetic simple.

Exhibits

Figure 1: Exhibit 1: A toy semantic map of the static embeddings

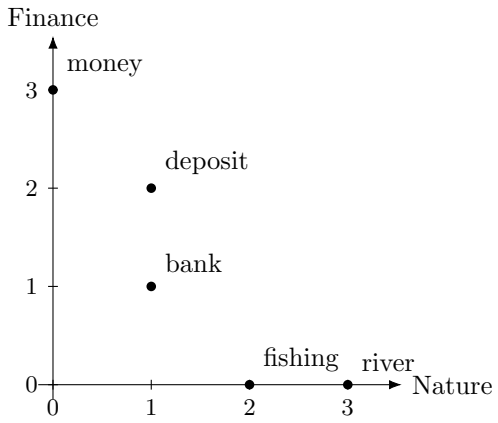

Table 1: Exhibit 2: Toy word embeddings in the two-dimensional semantic space

Word	**Static embedding [Nature, Finance]**	**Plain-English interpretation**
bank	$[1, 1]$	Ambiguous starting point: it could move toward a nature-related context or a finance-related context.
river	$[3, 0]$	Strongly nature-related.
fishing	$[2, 0]$	Nature-related.
money	$[0, 3]$	Strongly finance-related.
deposit	$[1, 2]$	Mostly finance-related, with some link to the target word *bank*.

Table 2: Exhibit 3: Softmax lookup for this exercise

If the scaled scores are ...	**Use these attention weights**
[1.5, 1.0, 1.0]	[0.45, 0.27, 0.27]
[1.5, 1.5, 1.0]	[0.38, 0.38, 0.23]

Table 3: Exhibit 4A: Worksheet for Context A — "river fishing bank"

Token	**Static embedding** x_i	**Similarity** $x_{\mathbf{bank}} \cdot x_i$	**Scaled score**	**Attention weight** w_i	**Weighted contribution** $w_i x_i$
river	$[3, 0]$				[,]
fishing	$[2, 0]$				[,]
bank	$[1, 1]$				[,]

$$y^{(A)}_{\text{bank}} = \sum_i w_i x_i = [\quad , \quad]$$

Table 4: Exhibit 4B: Worksheet for Context B — "money deposit bank"

Token	**Static embedding** x_i	**Similarity** $x_{\mathbf{bank}} \cdot x_i$	**Scaled score**	**Attention weight** w_i	**Weighted contribution** $w_i x_i$
money	$[0, 3]$				[,]
deposit	$[1, 2]$				[,]
bank	$[1, 1]$				[,]

$$y^{(B)}_{\text{bank}} = \sum_i w_i x_i = [\quad , \quad]$$

In-Class Questions

1. **Forward pass for Context A (do the mechanics).** Fill in Exhibit 4A for "river fishing bank." Work left-to-right: compute similarity, divide by 2, use the softmax lookup in Exhibit 3, then compute the weighted contributions and the final contextual embedding $y_{\text{bank}}^{(A)}$.

2. **Forward pass for Context B (repeat the same algorithm).** Fill in Exhibit 4B for "money deposit bank." Again work left-to-right: similarity → scaling → softmax → weighted sum.

3. **Interpret the two updated embeddings.** Compare $y_{\text{bank}}^{(A)}$ and $y_{\text{bank}}^{(B)}$.

   - In which context does *bank* move closer to the **Nature** region?
   - In which context does *bank* move closer to the **Finance** region?

   Explain in one or two sentences.

4. **Plot the movement on Exhibit 1.** Add the two new points for $y_{\text{bank}}^{(A)}$ and $y_{\text{bank}}^{(B)}$ to the semantic map in Exhibit 1. Draw arrows from the original point for *bank* to each new point. What does that movement show about **contextual embedding**?

5. **Complete the core ideas.**

   - Similarity is a _____ score based on the _____ between two embeddings.
   - Scaling keeps the scores from becoming too _____ before softmax.
   - Softmax converts scaled scores into _____ that add up to 1.
   - The **contextual embedding** is created by the _____ step.

6. **Business translation.** Suppose a model sees the word *charge* in customer text. Give two short contexts that would suggest different meanings of the word. Then explain, in one sentence, why self-attention would be more useful than a static embedding.