Attention Mechanism

What it does

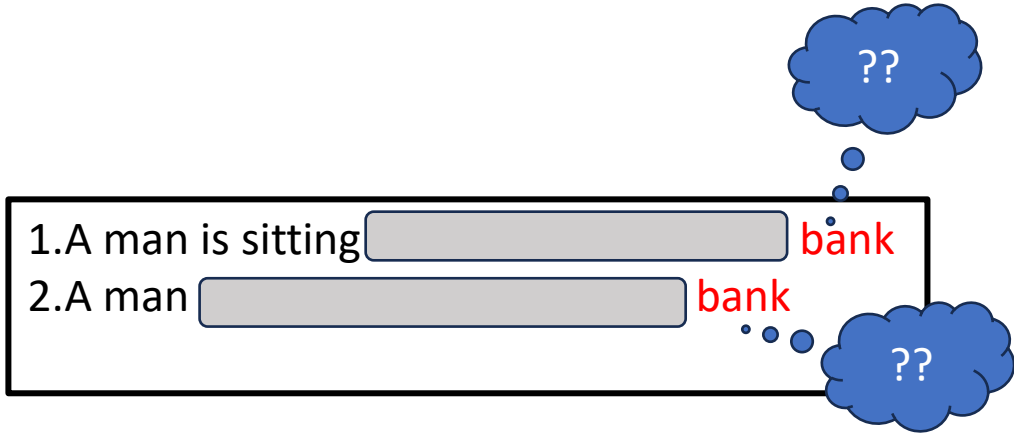
- Quantify how similar a word is to other words in a sentence (A sequence of words)

What happens afterwards

- It modifies the word embeddings by taking the context of the surroundings words

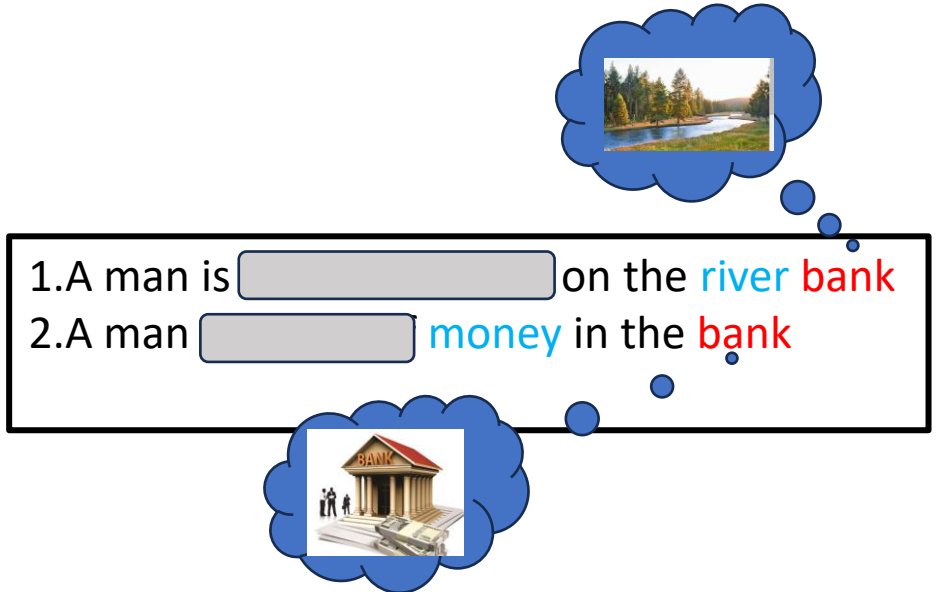
Why is it needed

To understand the context of each word and the overall sentence



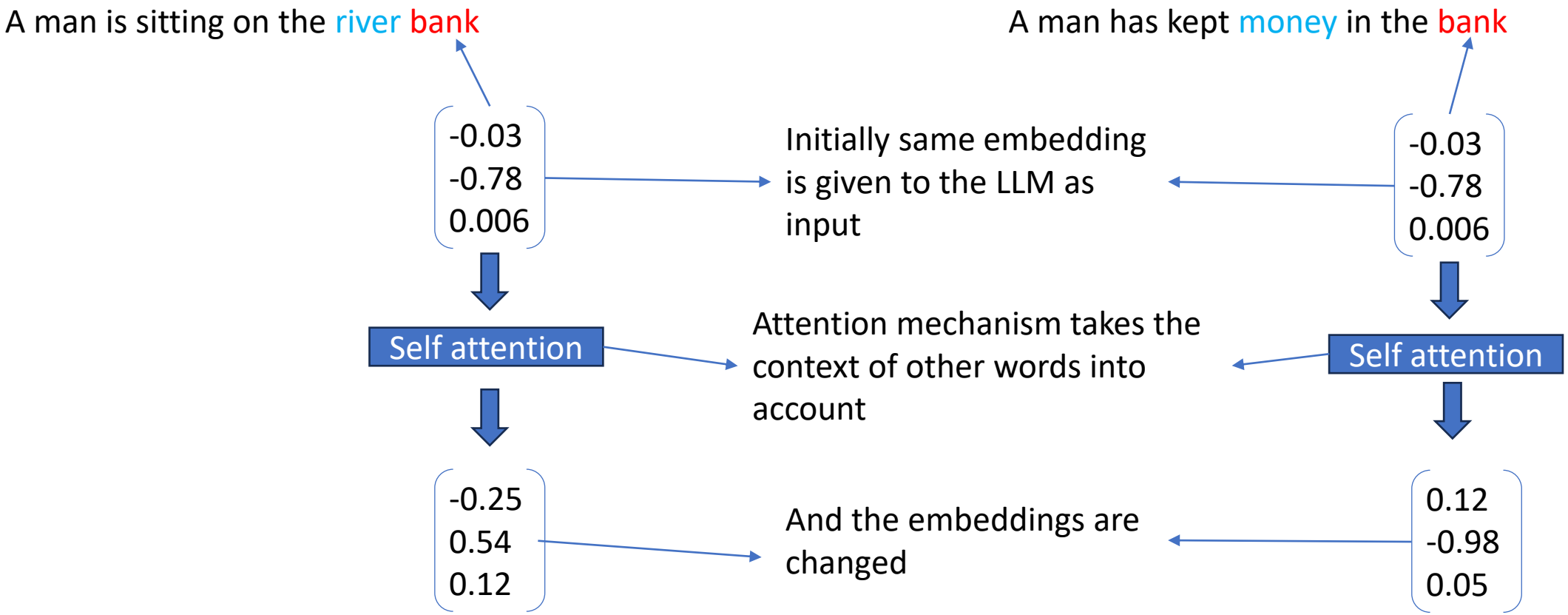
- To understand a sentence we need context of each word.
- We human being also can't infer the context or meaning of a word

  without referring to the other words
- If we are a given a word with multiple meanings we check the words before or after it to get some context
- In this example the word **"Bank"** has different meaning. Only after referring to the previous words **"river"** or **"money"** we get to know which bank is being talked about

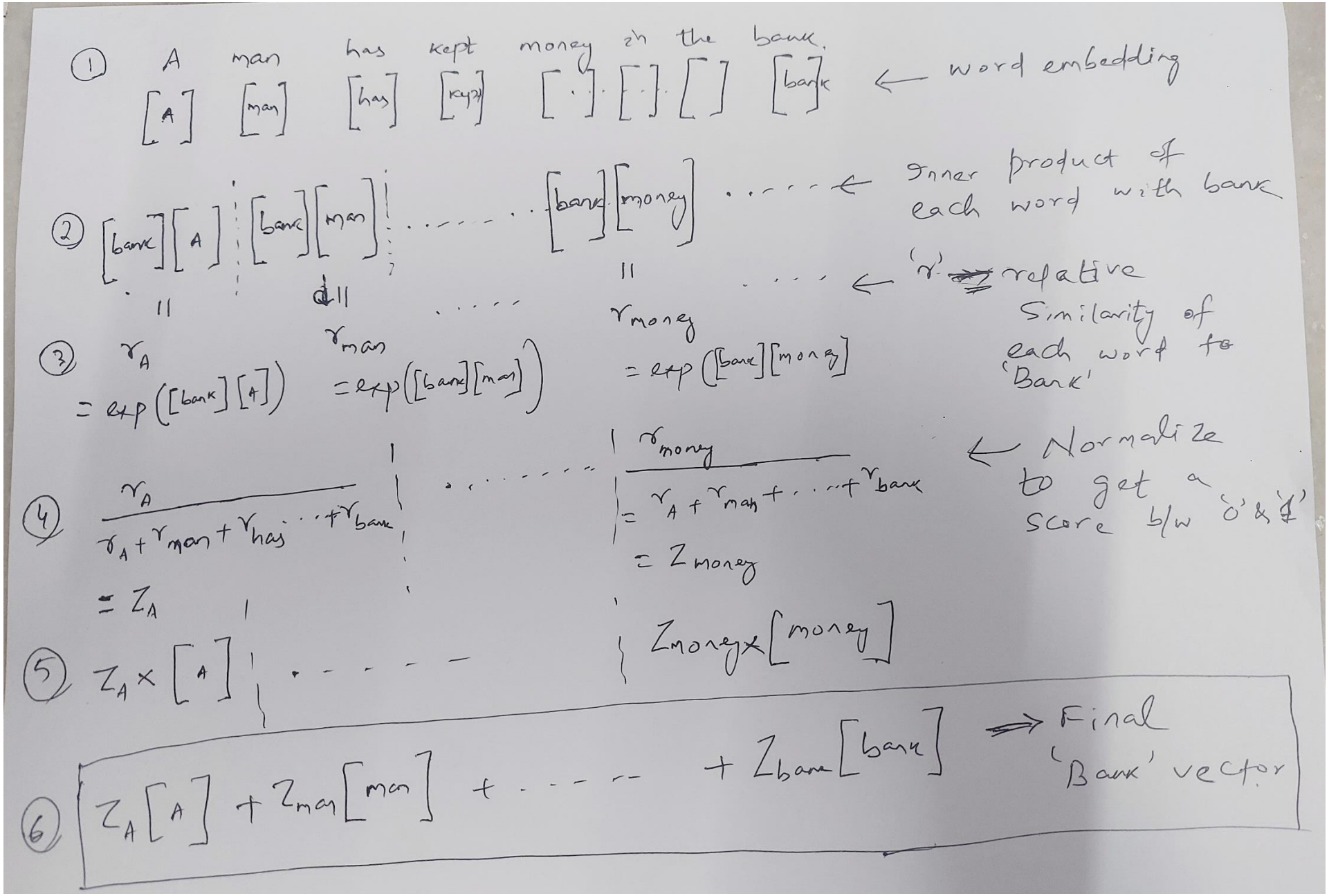


Attention Mechanism

- Same analogy can be applied to a LLM.
- A LLM is provided with the embedding vector of each word.
- How is it going to know the context or in this case which ‘bank’ we are talking about?
- It gets to know through this attention mechanism
- And accordingly the vectors are modified



Attention Math



Explanation of how a word ‘bank’ is paying attention to other word

For detailed understanding play around with the excel sheet

Attention Math

Assume each word has a 3-dimensional vector representation

		A		Man		has		kept		money		in		the		bank	Explaintion
		-0.03		-0.024		-0.148		-0.447		-0.207		-0.133		-0.013		0.02	Take each word's embedding
		-0.78		-0.259		-0.049		-0.265		-0.336		0.546		0.833		-0.286	
		0.006		-0.002		-0.242		-0.469		-0.411		0.076		-0.044		0.524	
		(A)dot(Bank)								(money)dot(Bank)							
		↓								↓							
		0.02		0.02		0.02		0.02		0.02		0.02		0.02		0.02	Attention quantifies how similar a word is to other words.In this case we are checking how similar the word "bank' is to other words
	'Bank'	-0.286		-0.286		-0.286		-0.286		-0.286		-0.286		-0.286		-0.286	
		0.524		0.524		0.524		0.524		0.524		0.524		0.524		0.524	
Inner Product		0.22562		0.07255		-0.11575		-0.17891		-0.12341		-0.11899		-0.26155		0.35677	We do that by taking inner product of ''Bank" with rest of the words.If two words are similar the inner product tends to be a positive number,if dissimilar the inner product is negative
Exponential	exp(inner product)	1.2531		1.07524		0.89069		0.83618		0.8839		0.88781		0.76985		1.42871	Now it is kind of inconvenient to deal with both positive and negative numbers. So we pass them through the exponential number to make them all positive
Normalizatior	r	0.15614		0.13398		0.11098		0.10419		0.11014		0.11062		0.09593		0.17802	Normalize the exponential by dividing the sum of all exponential.It represents the relative similarity and adds to 1. If the value is more towards 1 , then the word "bank" is strongly related to that particular word and vice versa as it is a number between 0 & 1
		-0.00468		-0.00322		-0.01643		-0.04657		-0.0228		-0.01471		-0.00125		0.00356	r' is then multiplied with each word vector.'r' acts as a weight of the word vector.If 'r' is very strong , when it multiplies with the corresponding word it is given more weightage than other words.
	(r) x(each word)	-0.12179	+	-0.0347	+	-0.00544	+	-0.02761	+	-0.03701	+	0.0604	+	0.07991	+	-0.05091	
		0.00094		-0.00027		-0.02686		-0.04887		-0.04527		0.00841		-0.00422		0.09328	
	Modified 'Bank' vector			-0.1061													All the vectors are added together to get the final modified vector.Now this modified vector contains the context from all other words. If a word is highly related to the word 'Bank', that will contribute significantly to the final modified word vector of 'Bank' or in other words the word 'bank' will contain more information from the highly related word 'money'
				-0.13715													
				-0.02285													