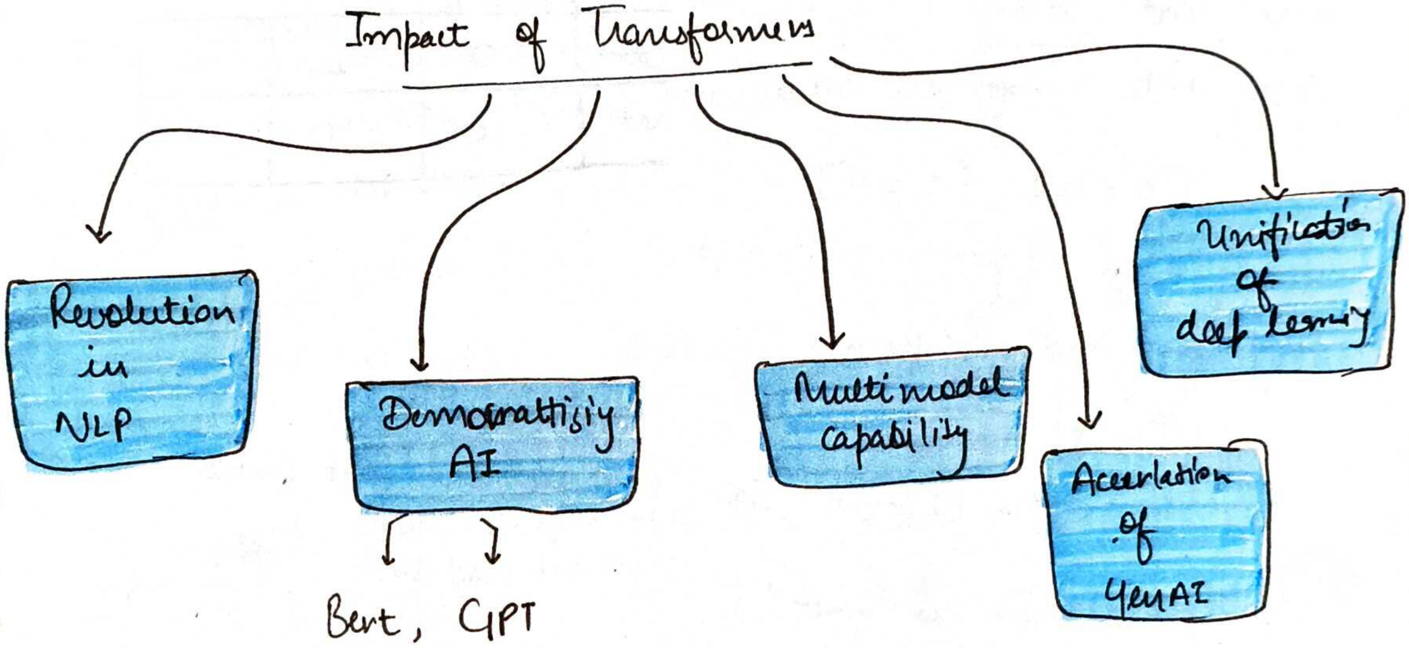
Transformers

ANN → Tabular Data

CNN → Image Data

RNN → Sequence data (Text)

Transformer → Sequence to Sequence task



Self Attention

The what

NLP → $\underbrace{\text{Words to number}}_{\text{Very important}}$ [vectorization]

Methods.

1. OHE

mat cat mat

Cat rat rat

[1 0 0] [0 1 0] [1 0 0]

	mat	cat	rat
mat	1	0	0
cat	0	1	0
rat	0	0	1

2) Bag of Words (BOW)

	mat	rat	cat
S_1	2	0	1
S_2	0	2	1

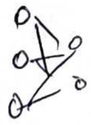
1 → Cat 1 times in first sentence

↳ mat 2 times in first sentence

S_2 ↳ 2nd sentence

3) Word embedding → Semantic Meaning

Training Data (Very large) send →



→ $\begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$ ↳ n-dim vector

each word converted into n-dim vector

let say, we have 5-dim vector

king → [0.6 0.2 1.0 0.9]

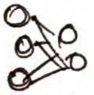
queen → [0.3 0.2 0.4 1.0].

If both words are similar or same than Vector of the both words will be similar because of we are finding Semantic.

The problem of "Average Meaning"

1) An apple a day keeps the doctor away

2) Apple is healthy.

3) Apple is better than orange. → [ ]  → [:]

2 dim

4) Apple makes great phones

[X Y]

taste ↲ ↳ technology

⋮

1st line → Apple khane/taste → ki baat ho rhi [X Y] ↓ ↓ [0.6 0]

2nd line → More sure ↓ Apple khane/taste ki baat ho rhi [0.7 0]

3rd line → Apple khane/taste ki baat ho rhi [0.8 0]

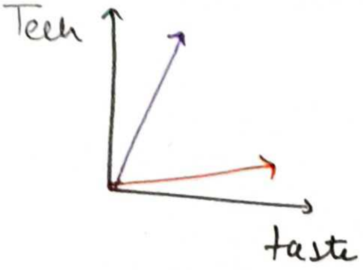
4th line → Technology ki bat ho rhi [0.8 0.2]

Total sentence → 10000

9000 sentence ↓ Fruits/taste

1000 sentence ↓ Tech

→ Overall vector [0.9 0.3] more tilted to taste than tech



Data → tilted toward taste than Tech

Data → tilted toward tech than taste

Problem

Word embedding create one time and use many time.

↳ Static

↳ ek baar embedding ban gya to bar bar wahi use karn hai.

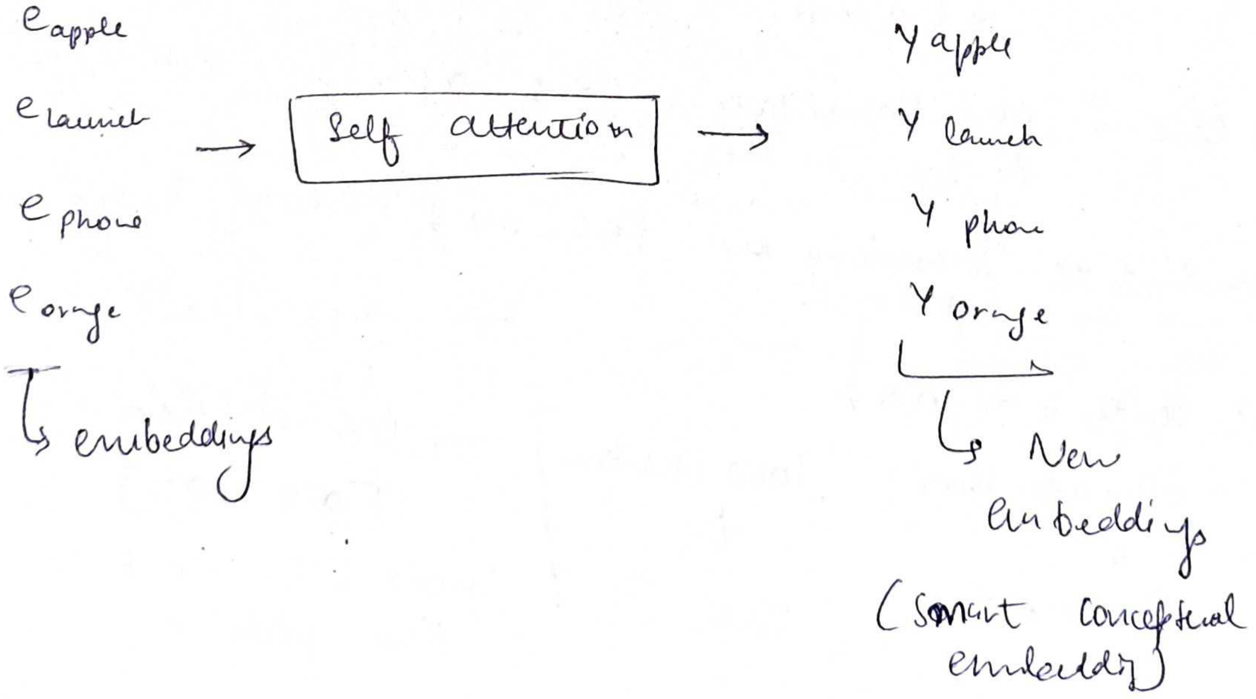
eg: Apple launched a new phone while I was eating an orange.

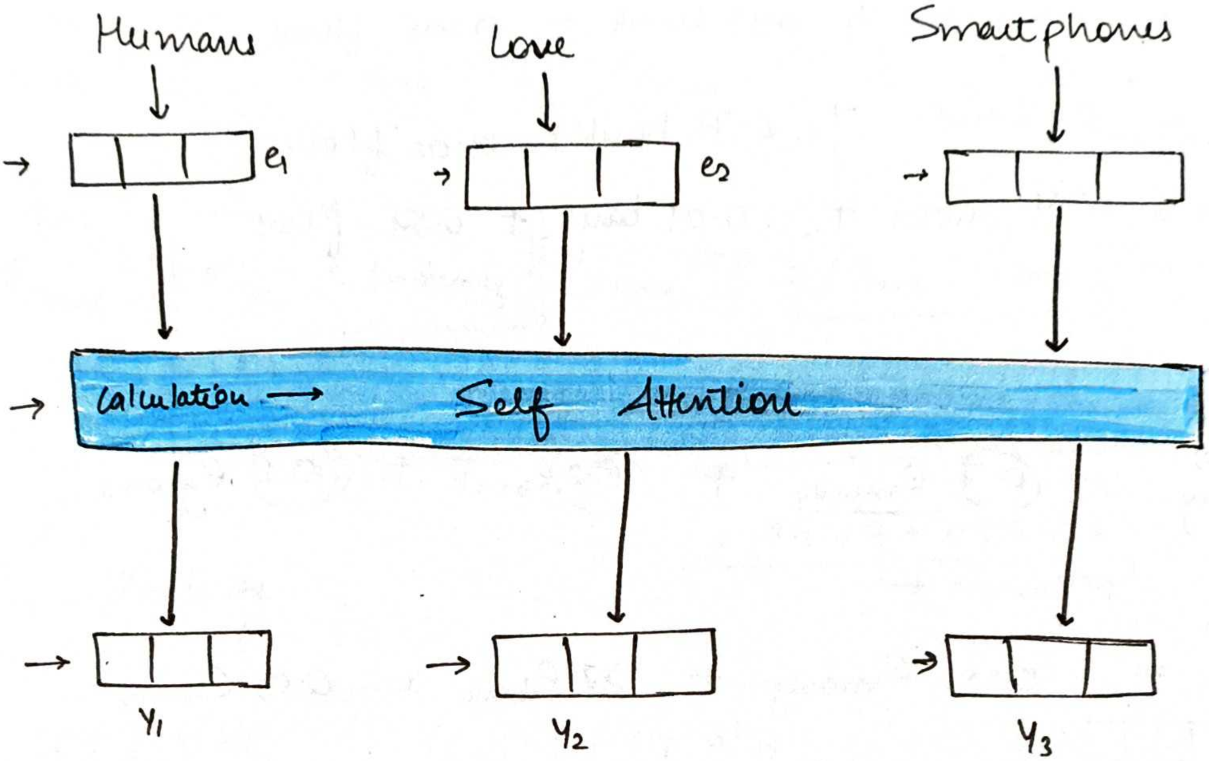
↳ Eng to hindi Translation

* But our data is tilted toward taste So this Apple treated as a fruit not technology. [0.9 0.3]

But we want to change value based on content.

So, this problem solve self Attention.





First principle Approach

money bank grows

bank → 0.3 money + 0.7 bank + 0.1 grows

so, bank word not only made with bank but also made with other words too

rivers bank flows

bank → 0.5 river + 0.4 bank + 0.1 flows

money = 0.7 money + 0.2 bank + 0.1 grows

bank = 0.25 money + 0.7 bank + 0.05 grow

grows = 0.1 money + 0.2 bank + 0.7 grows river = 0.8 river + 0.15 bank + 0.05 flows

bank = 0.2 river + 0.78 bank + 0.02 flows

flows = 0.4 river + 0.01 bank + 0.59 flows

$$e_{money}^{(new)} = \boxed{0.7}\, \overset{\text{n dim vector } \square\square\square}{\underline{e_{money}}} + \boxed{0.2}\, e_{bank} + \boxed{0.1}\, e_{grows}$$

$$e_{bank}^{(new)} = 0.25\, e_{money} + 0.7\, e_{bank} + 0.05\, e_{grows}$$

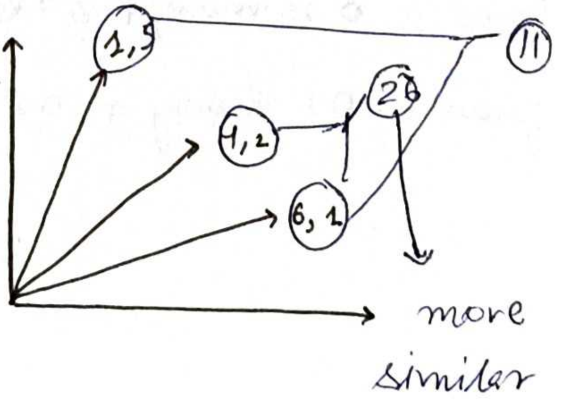
$$e_{grow}^{(new)} = 0.1\, e_{money} + 0.2\, e_{bank} + 0.7\, e_{grows}$$

0.7 times old embedding money se bna hai
0.2 times old embedding bank se bna hai
0.1 time old embedding grows se bna hai

or we can say

0.7 → similarity betn money embedding and money embedding
0.2 → similarity betn money embedding and bank embedding
0.1 → similarity betn money embedding and grows embedding

And All these embedding are vector. And Dot product betn vectors is known as similarity

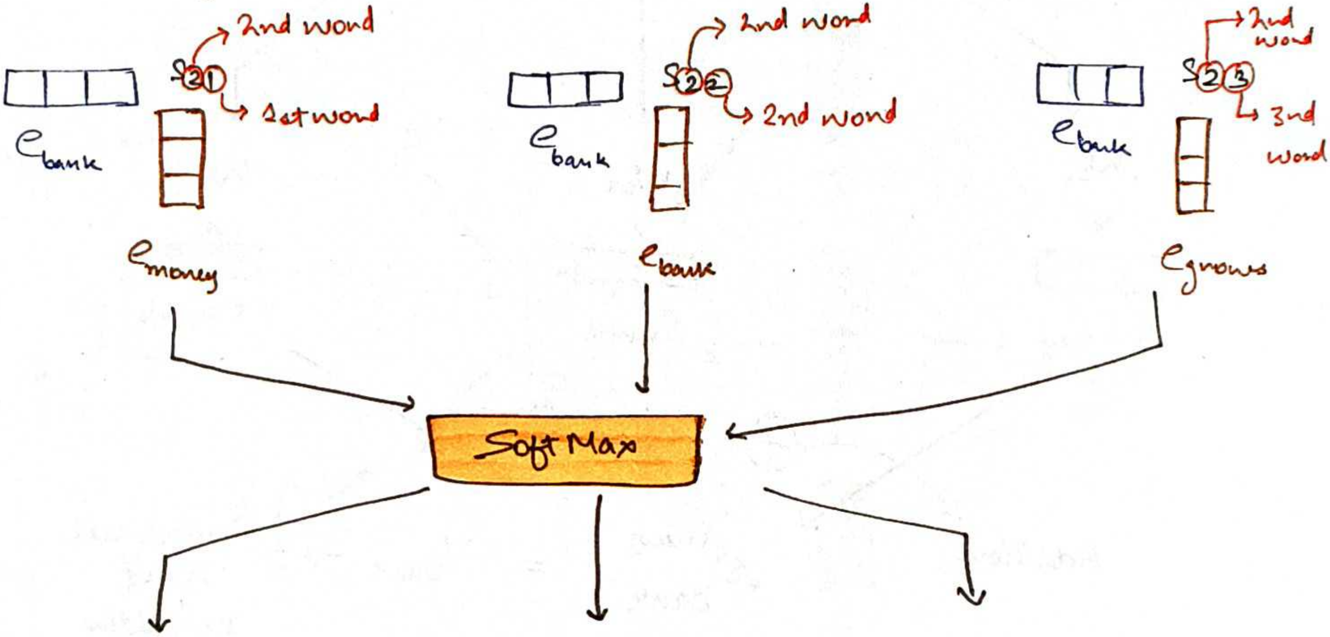


$$e^{(new)}_{bank} = 0.25\, e_{money} + 0.7\, e_{bank} + 0.05\, e_{grows}$$

$$e^{(new)}_{bank} = \boxed{[e_{bank} \cdot e^{T}_{money}]}\, e_{money} + \boxed{[e_{bank} \cdot e^{T}_{bank}]}\, e_{bank} + \boxed{[e_{bank} \cdot e^{T}_{grows}]}\, e_{grows}$$

Similarity

$\underbrace{0.25 + 0.7 + 0.05}_{} = 1$ ↳ Normalized



W_{21}

$$W_{21} = \frac{e^{s_{21}}}{e^{s_{21}} + e^{s_{22}} + e^{s_{23}}}$$

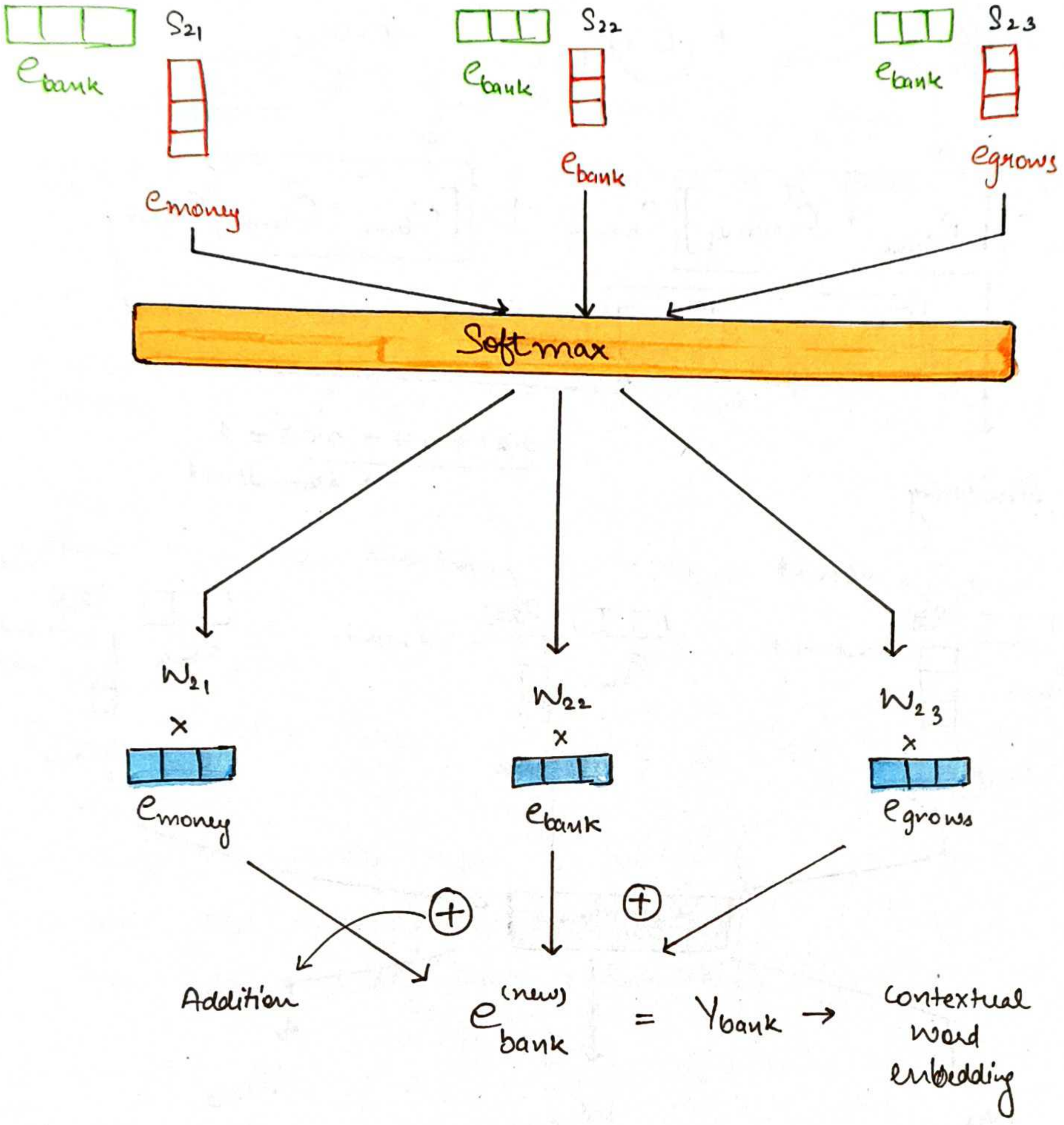
W_{22}

$$W_{22} = \frac{e^{s_{22}}}{e^{s_{21}} + e^{s_{22}} + e^{s_{23}}}$$

W_{23}

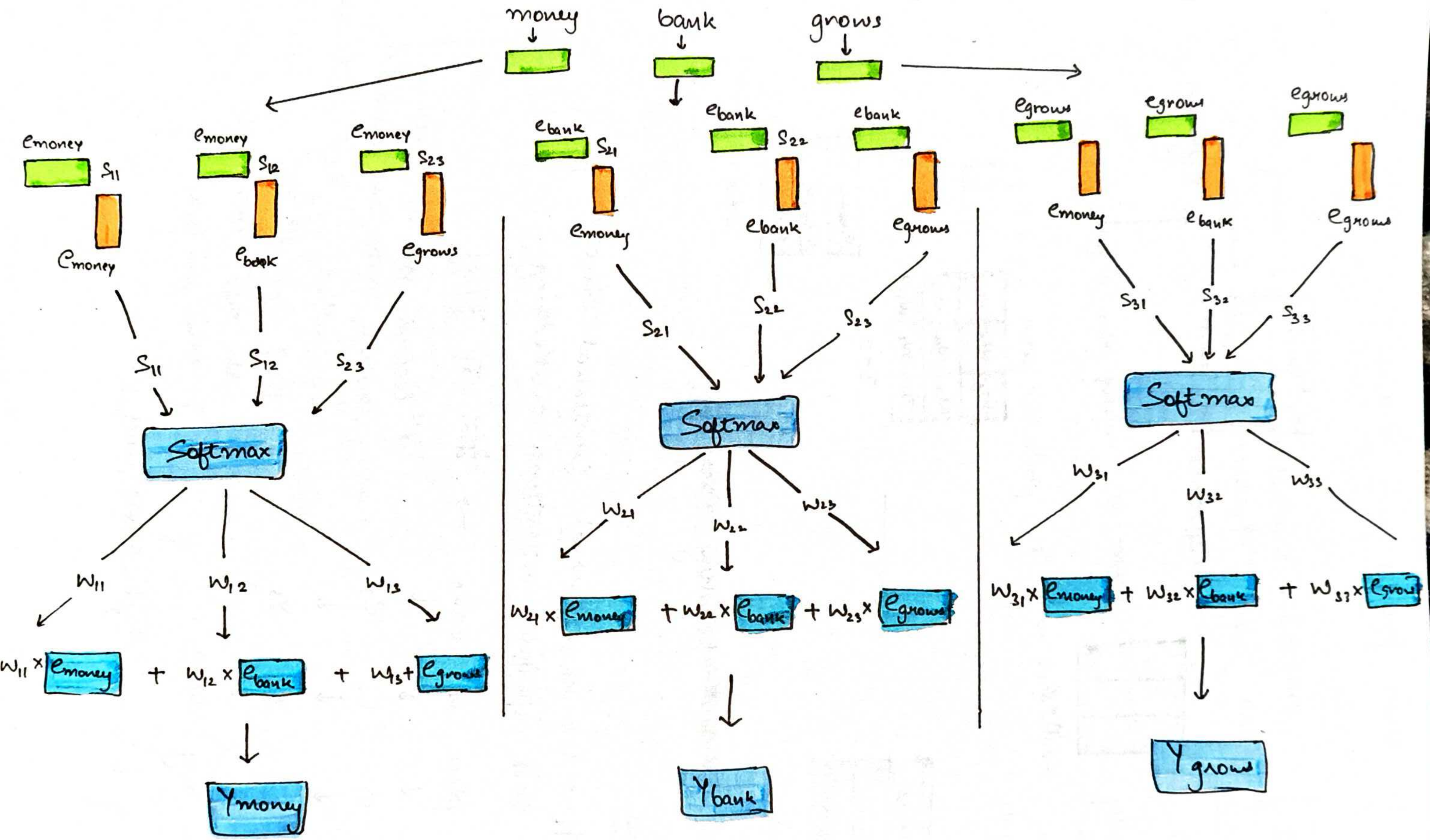
$$W_{23} = \frac{e^{s_{23}}}{e^{s_{21}} + e^{s_{22}} + e^{s_{23}}}$$

W_{21} , W_{23} and W_{22} is Normalized because sum of $W_{21} + W_{23} + W_{22}$ is 1.

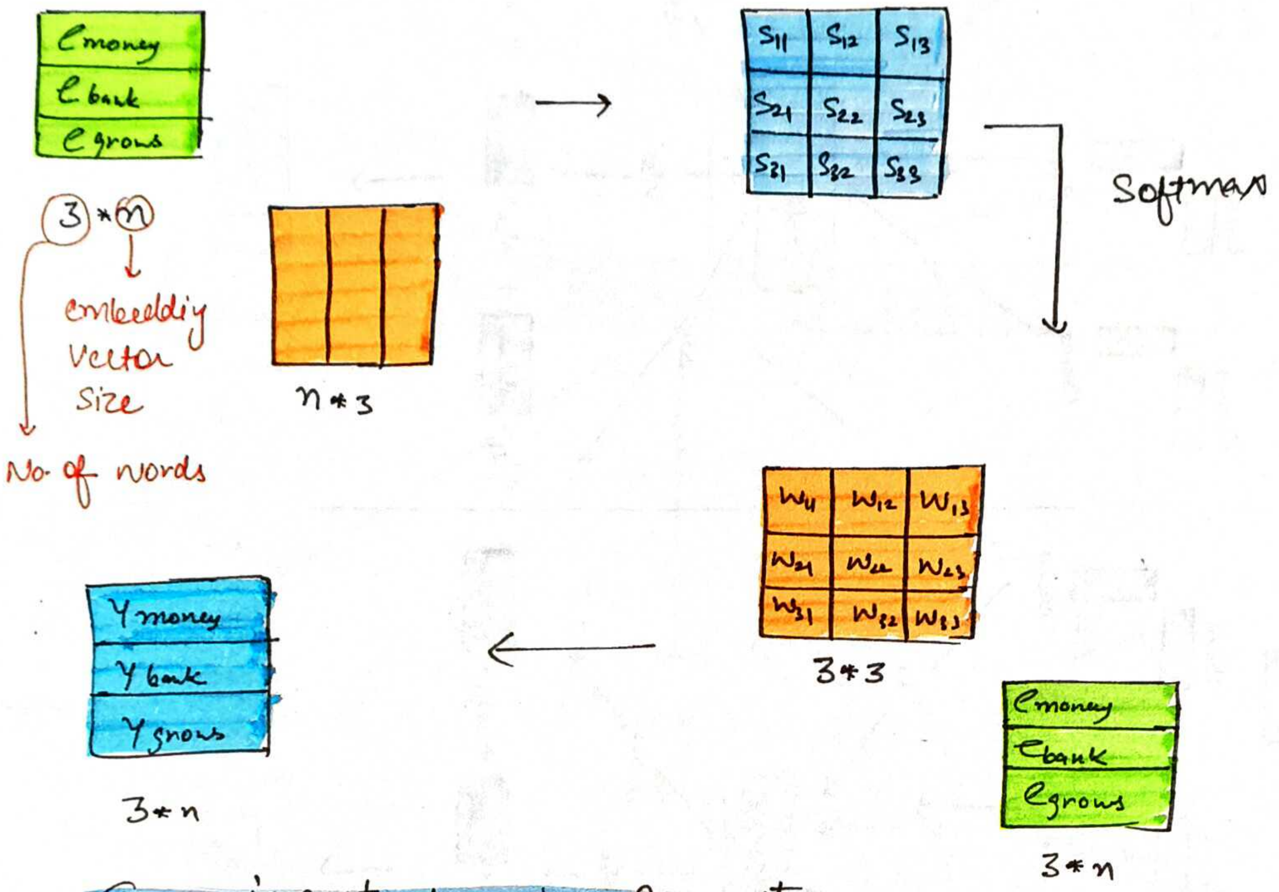


This diagram is only for Y_{bank}

and Y_{bank} is contextual word embeddig


money
bank
grows
emoney
S11
emoney
emoney
S12
ebook
emoney
S23
egrows
S11
S12
S23
Softmax
W11
W12
W13
W11 × emoney + W12 × ebank + W13 + egrows
Ymoney
ebank
S21
emoney
ebank
S22
ebank
ebank
egrows
S21
S22
S23
Softmax
W21
W22
W23
W21 × emoney + W22 × ebank + W23 × egrows
Ybank
egrows
emoney
egrows
ebank
egrows
egrows
S31
S32
S33
Softmax
W31
W32
W33
W31 × emoney + W32 × ebank + W33 × egrows
Ygrows

How to these process in Parallel ?



→ There is not Learning Parameter.

Problem

Our Approach is general contexual embedding Not task specific contentual embedding.

eg:- piece of cake → केक का टुकड़ा

$$\begin{tikzcd} piece \arrow[d] & of \arrow[d] & cake \arrow[d] \\ e_1 \arrow[d] & e_2 \arrow[d] & e_3 \arrow[d] \\ y_1 & y_2 & y_3 \end{tikzcd}$$

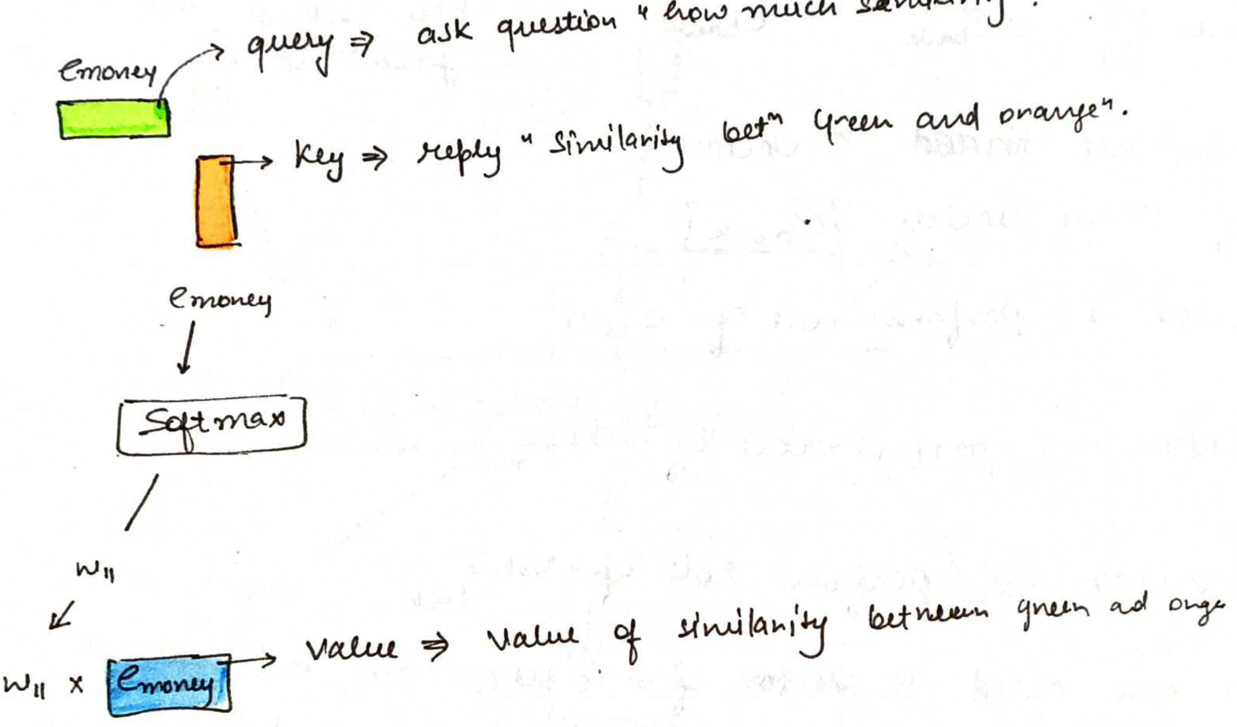
* Because we are using general contentual embedding our output is ~~बहुत / आसान काम~~- ~~Kabhi~~ "केक का टुकड़ा. Kabhi bhi "बहुत आसान काम" output nhi kar sakta.

but in my data

piece of cake → बहुत आसानकाम

If Task specific contextual embedding use then might be output is "बहुत आसान काम".

At some point general contextual embedding will fail. If I am doing sentiment analysis then embedding → according to sentiment analysis task.



Emoney → query ⇒ ask question "how much similarity".

→ key ⇒ reply "similarity betn green and orange".

Emoney → Softmax → w_{11}

w_{11} × Emoney → value ⇒ value of similarity between green and orange

d= {a:2, b:3, c:4} → dictionary

value

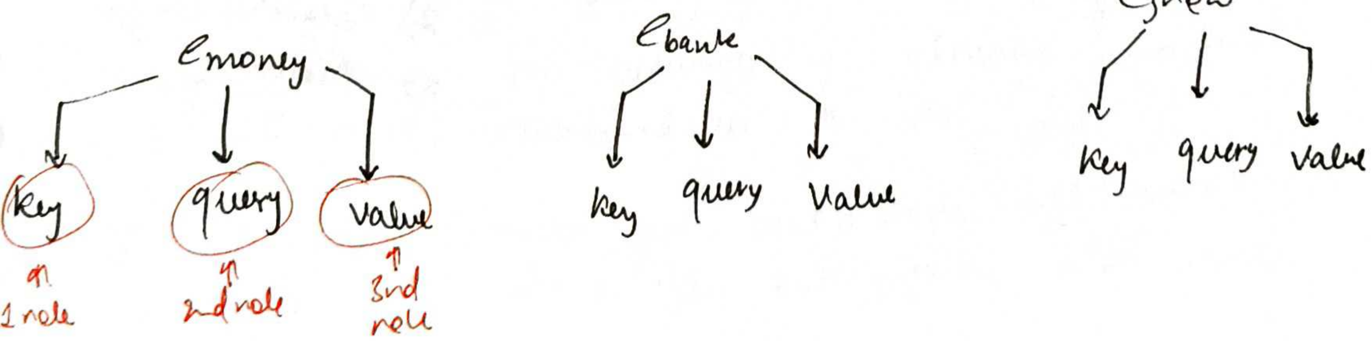
Key

d[a] = ?

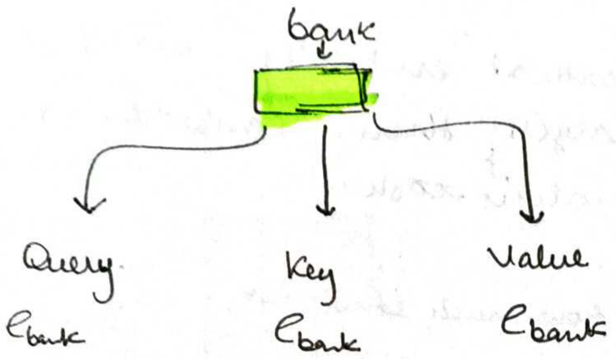
↳ query

→ doing same thing on diagram.

→ every embedding play 3 roles.



Query , Key & Value Vectors



Problem

Ye query bhi khud hi ban jata hai. Key bhi and Value bhi khud hi ban jata hai.

* ideally we need 3 vector of this vector. $\boxed{e_{bank}}$

1 Vector → perform role of $query_{bank}$

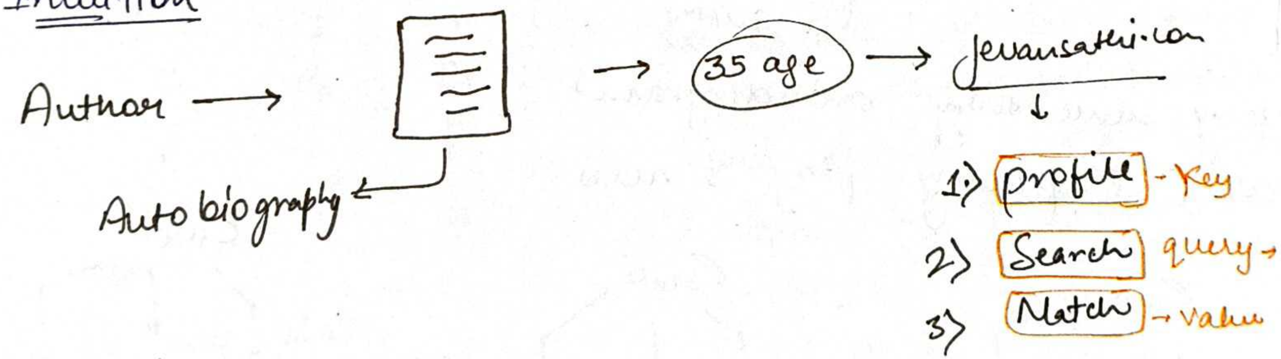
2 Vector → perfone role of Key_{bank}

3 Vector → perfom role of $Value_{bank}$

Why we need 3 vector for query, key ad value?

because single vector cannot work like query, key and value. That is why create particular vector for each query, key and value.

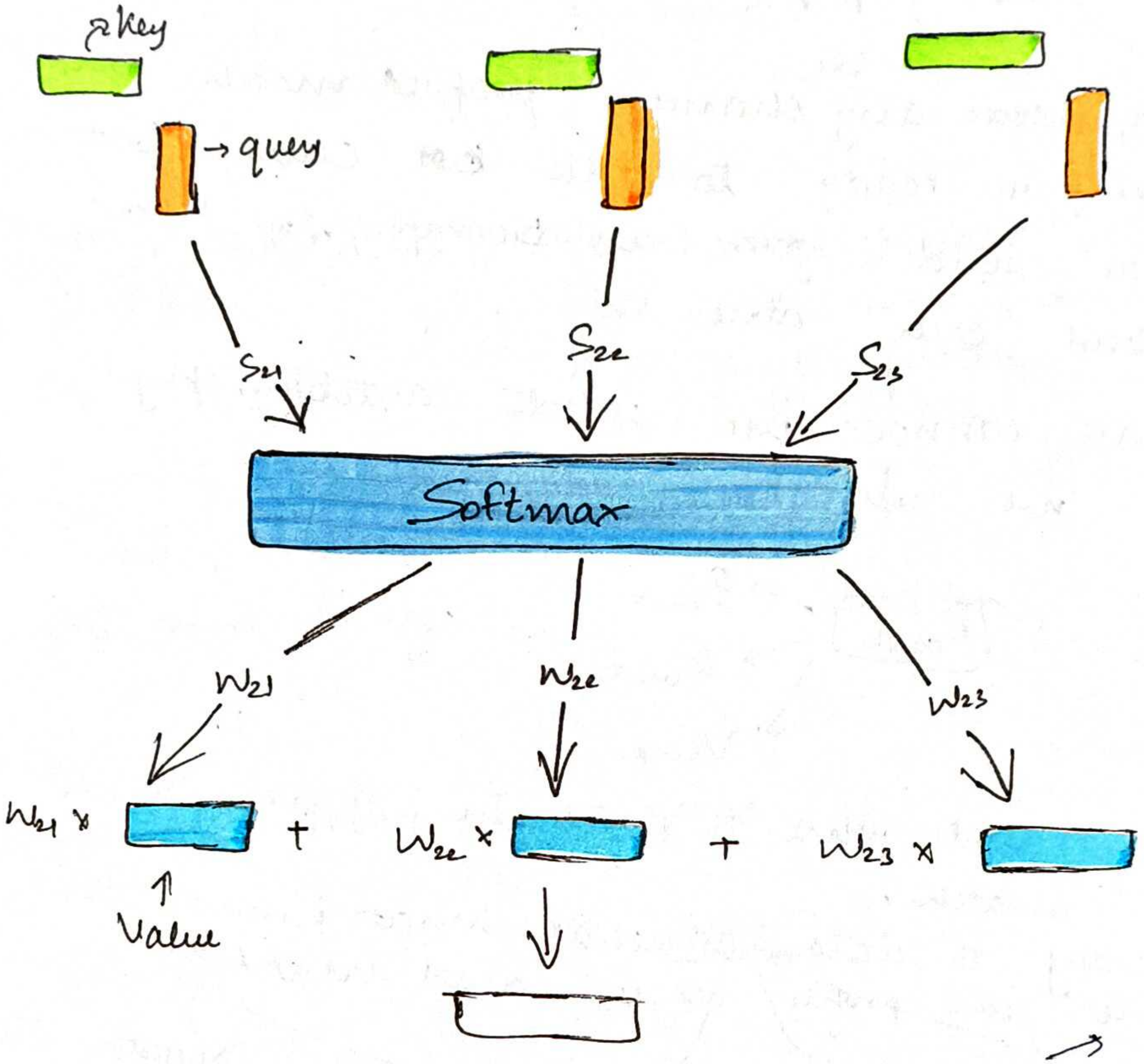
Intuition



query → qualification of girl, state of living, hobbies

Key → My profile ⇒ girl know about me .

Value → After match ⇒ Start conversation.



Embedding → key
→ query
→ value

Conclusion of this example:

1. If I want to share my profile to girl
then I'll share personalities, good things, hobbies.
If I'll shore my Autobiography instead of simple
profile because in autobiography already written so,
it will create awkwardness to other girl.

2. If I want search girl and I write all autobiography
in search bar because In autobiography also wrote

about which type of girls I like to marry. Wrote whole biography in search bar and may be ~~give~~ suggest wrong girls option because of long autobiography.

3. By ~~chance~~ any chance profiles match and Girl is ready to talk ~~or~~ chat but you send your autobiography to know about each other.

So, we cannot use whole autobiography. That's we use key, query, value.

$$\begin{tikzcd}
 & q_{bank} \\
\boxed{e_{bank}} \arrow[ur] \arrow[r] \arrow[dr] & k_{bank} \\
 & V_{bank}
\end{tikzcd}$$

How I decide what to write in Profile, Search and match.

→ According to data I decide what i want to write in profile search and match.

How!

→ First I write in profile → I wrote political books $\xrightarrow{\text{Suggestion}}$ Those ~~only~~ girls who are interested in politics

but I don't like girls who are interested in politics ← Those girls who are interested in politics

According to data, I understood that I have to change profile from Political book writer to Authors / writer ← but I don't like girls who are interested in politics

→ First I write in Search → Working proffesion / Non-Working

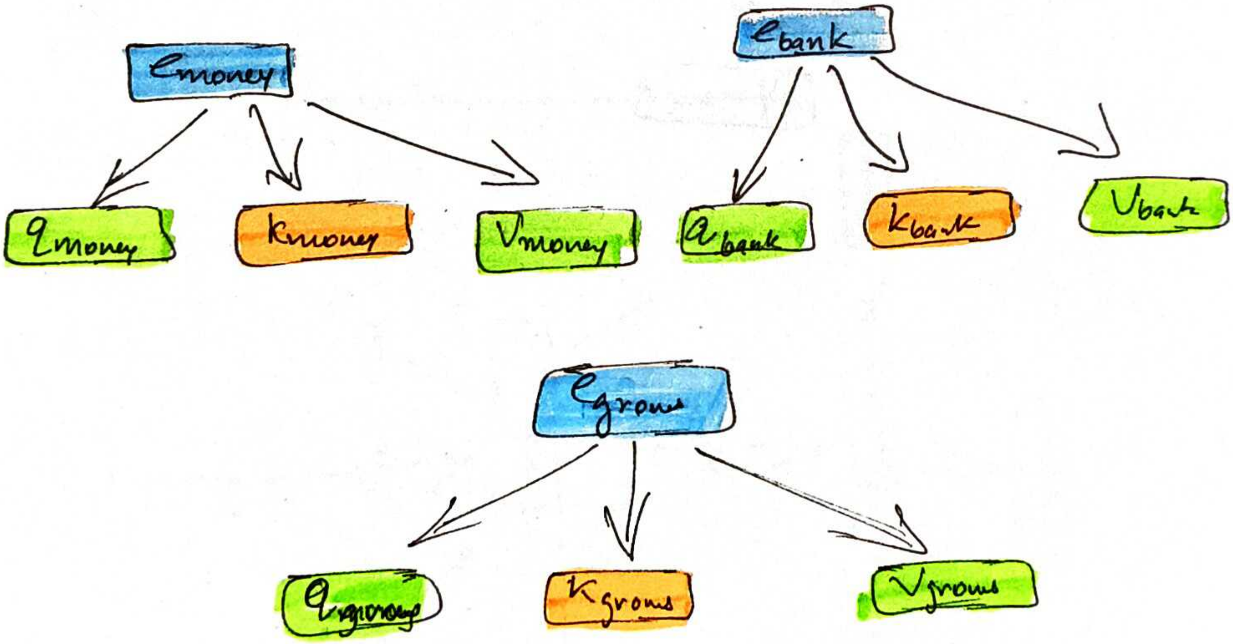
After conversation with of both type of girls — Data

Decision ← I like only working girls

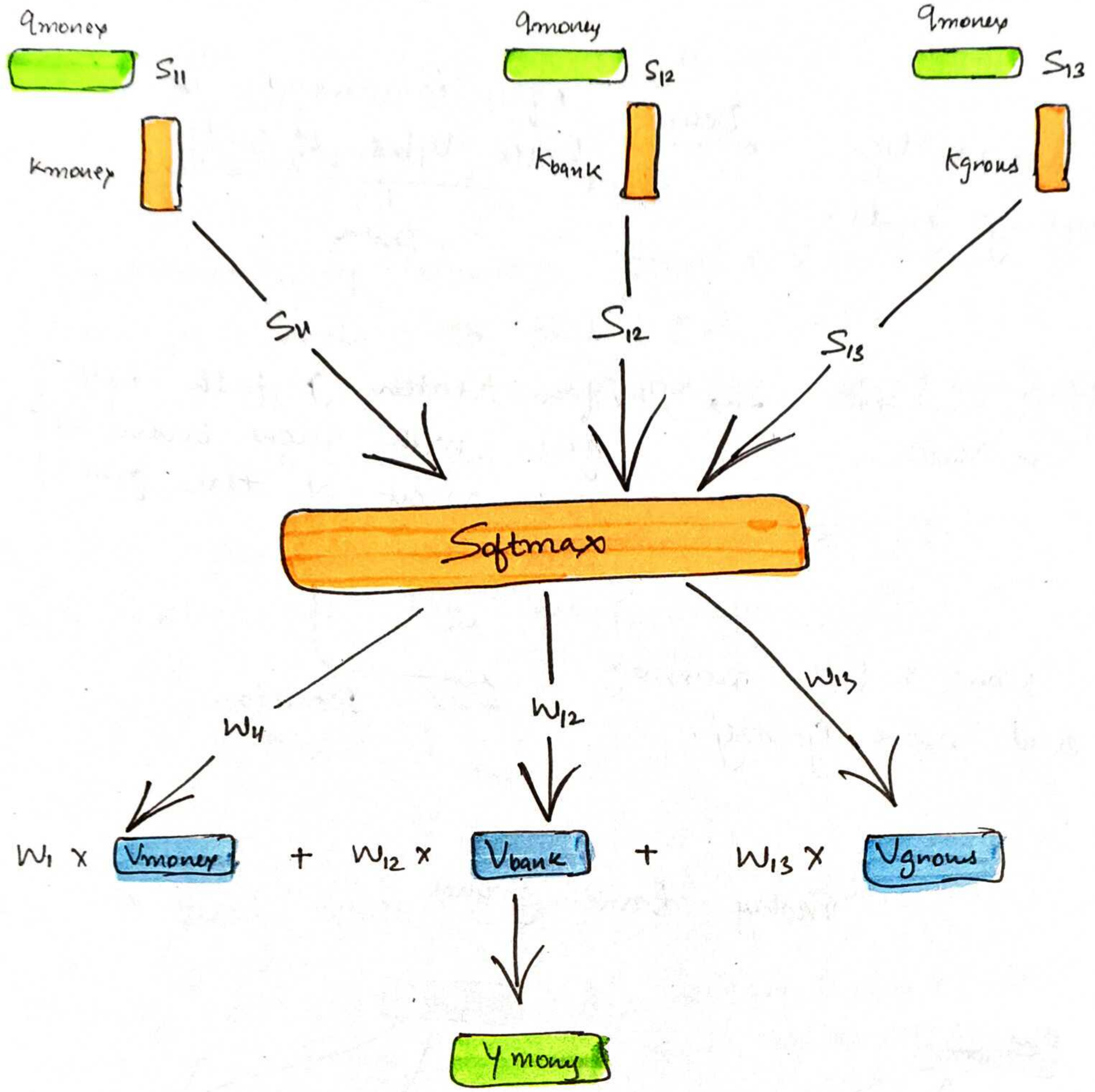
→ First I write in Match → After Match I talk with girls with more enthusiast And most of the girl not like it. — Data

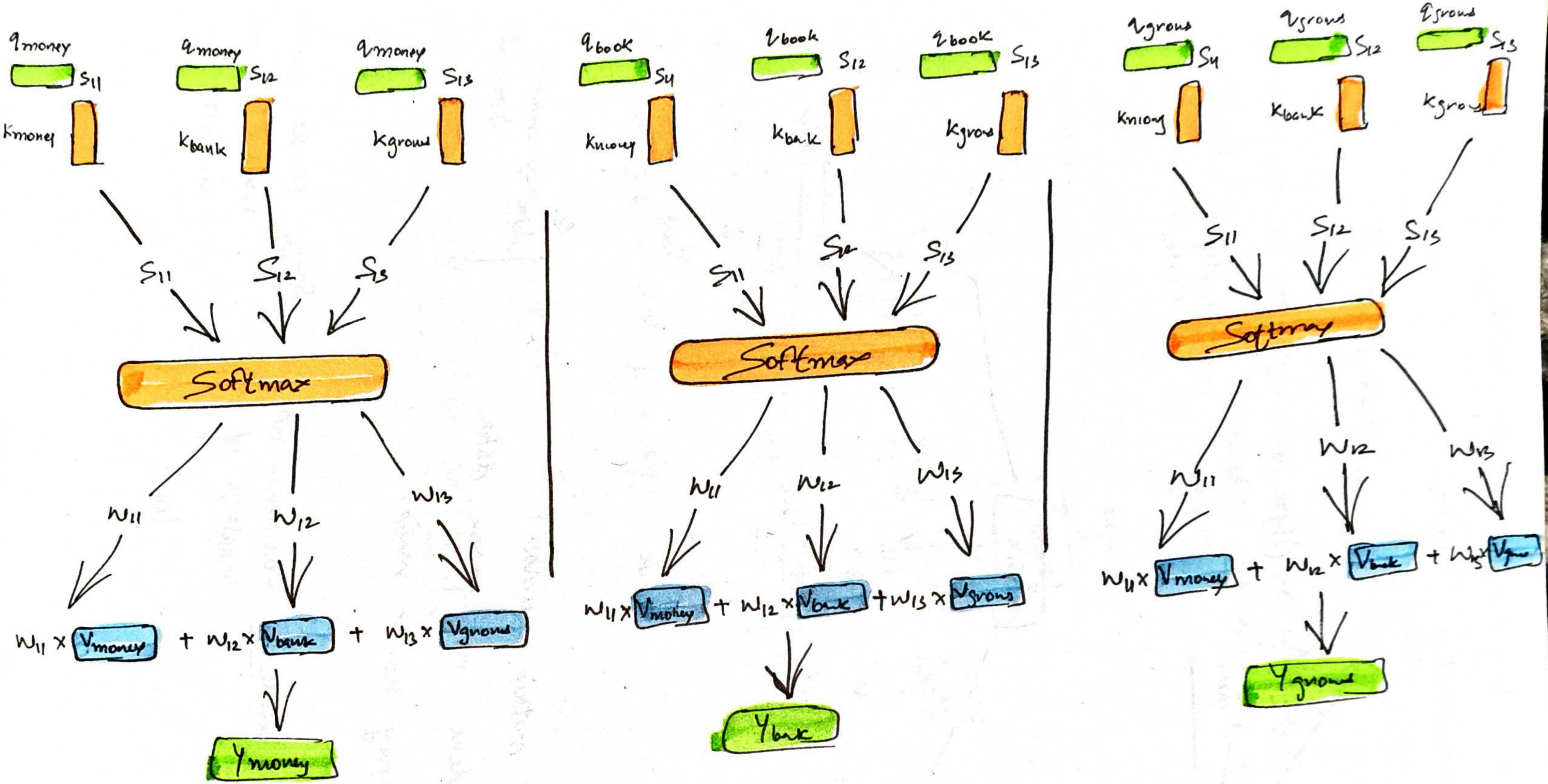
Decision → Now, I talk normaly and more Gently.

money bank grows



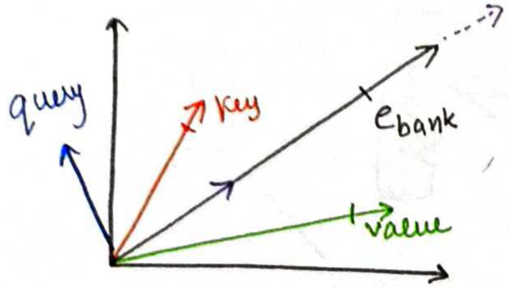
Money


qmoney
S11
kmoney
qmoney
S12
kbank
qmoney
S13
kgrows
S11
S12
S13
Softmax
W11
W12
W13
W1 x
Vmoney
+
W12 x
Vbank
+
W13 x
Vgrows
Y mony



qmoney
S11
Kmoney
qmoney
S12
Kbank
qmoney
S13
Kground
S11
S12
S13
Softmax
W11
W12
W13
W11 × Vmoney + W12 × Vbank + W13 × Vground
Ymoney
qbook
S11
Kmoney
qbook
S12
Kbank
qbook
S13
Kground
S11
S12
S13
Softmax
W11
W12
W13
W11 × Vmoney + W12 × Vbank + W13 × Vground
Ybank
qground
S11
Kmoney
qground
S12
Kbank
qground
S13
Kground
S11
S12
S13
Softmax
W11
W12
W13
W11 × Vmoney + W12 × Vbank + W13 × Vground
Yground

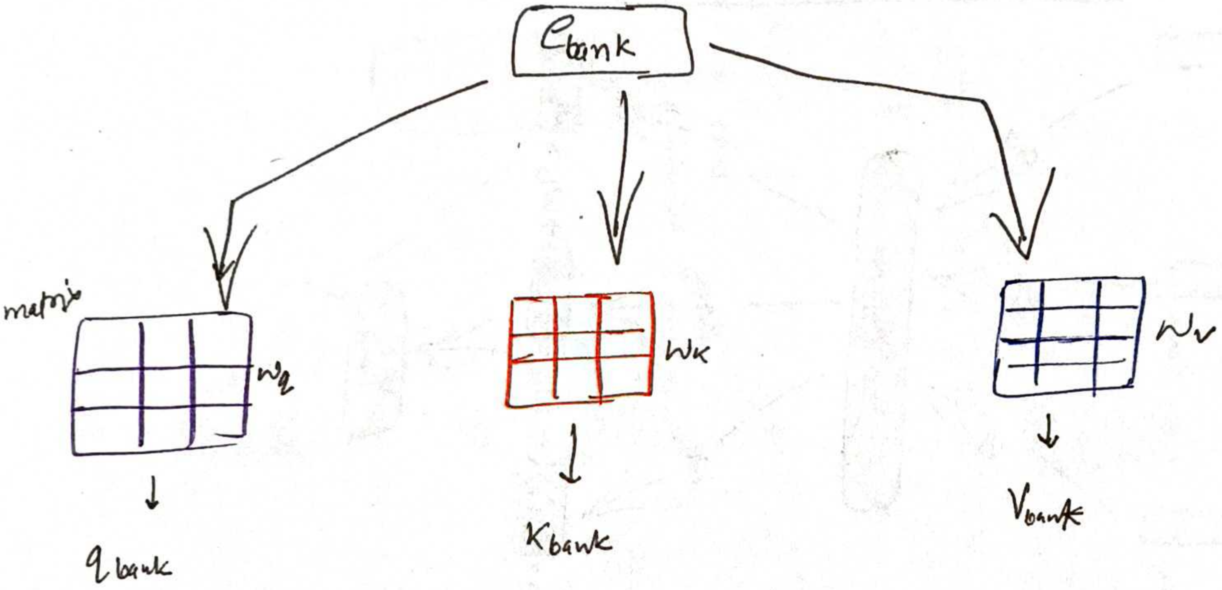
How to make key vector, query vector, value vector from embedding vector.



(i) magnitude (Scaling)

(ii) Linear transform

→ Multiply with Matrix



How to find matrix number?

→ first random no. from data

→ Then improving no. in matrix.

eg:
data → machine translation

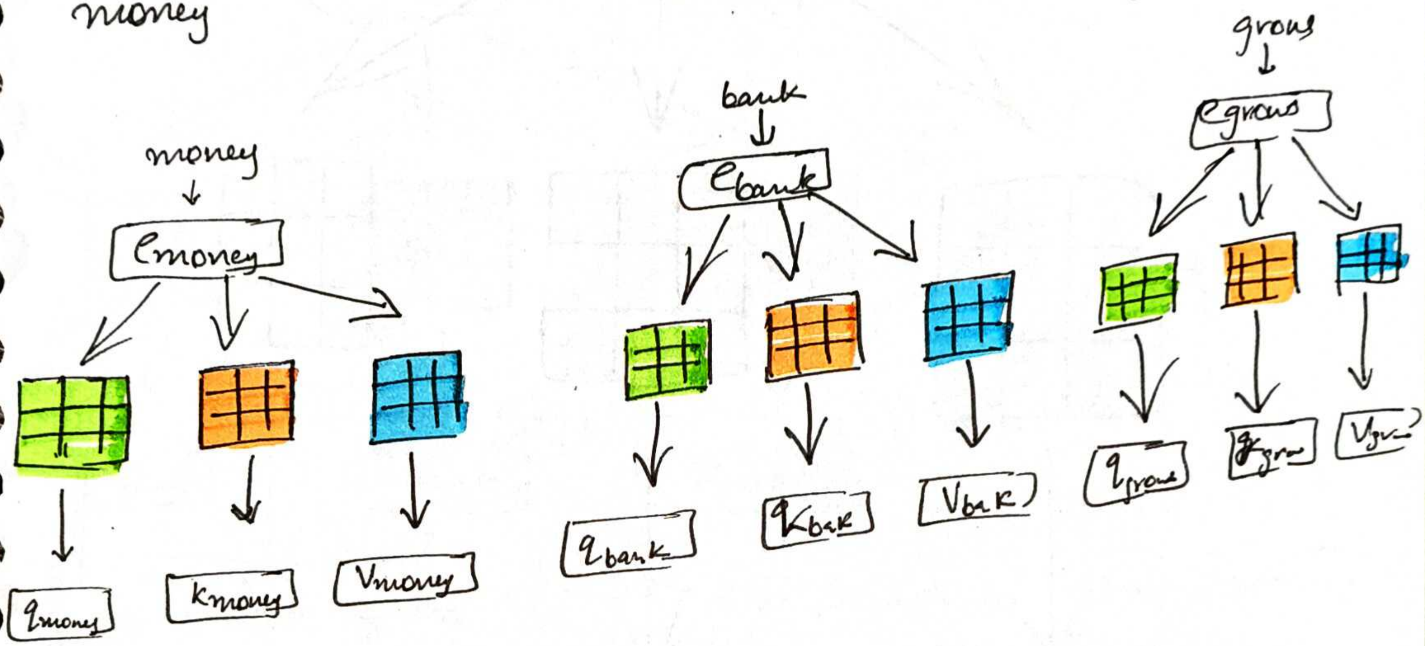
random no. in matrix of wq
Money

$\xrightarrow{\text{same}}$ random no. in maatrix of wq
bank

$\xrightarrow{\text{Same}}$ random no. in matrix of wq
in grens

random no. of matrix in W_k (money) $\xrightarrow{\text{Same}}$ random no. of matrix in W_k (bank) $\xrightarrow{\text{Same}}$ random no. of matrix in W_k (grows)

random no. of matrix in W_v money $\xrightarrow{\text{Same}}$ random no. of matrix in W_v bank $\xrightarrow{\text{Same}}$ random no. of matrix in W_v grows



All this process are parallely.

Mathematical formula

$$\boxed{\text{Attention }(QK,V) = \text{Softmax}\,(QK^T)V}$$