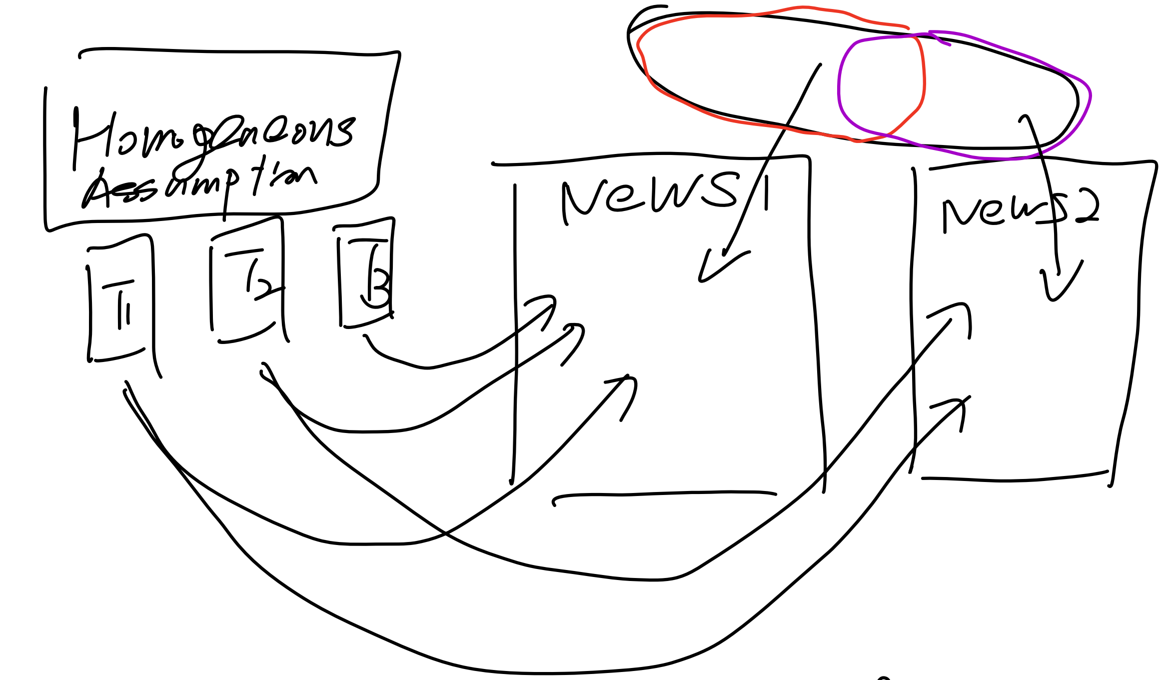
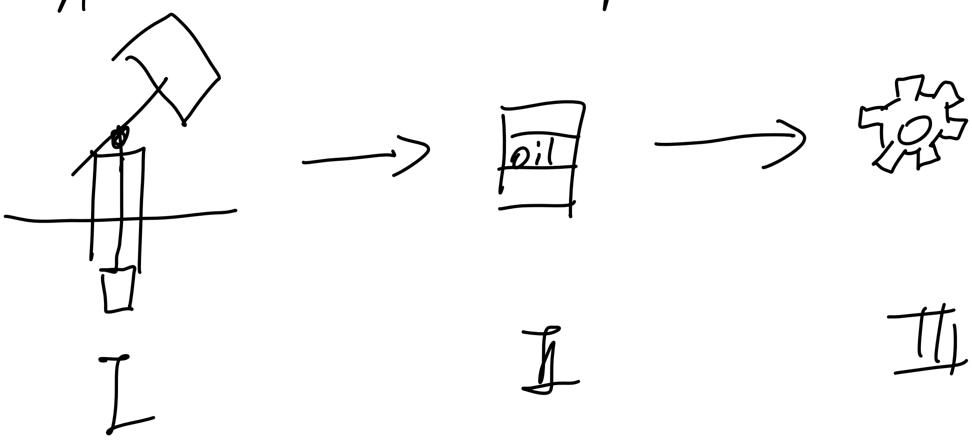


Bayesian classifier:

Need to Classify Good & Bad for inc A.

A is a crude oil production inc.



Text I

oil has release

new tech on

oil extraction

Text II

WB bought 100M to

oil Inc. . . .

making it fast
and high-quality

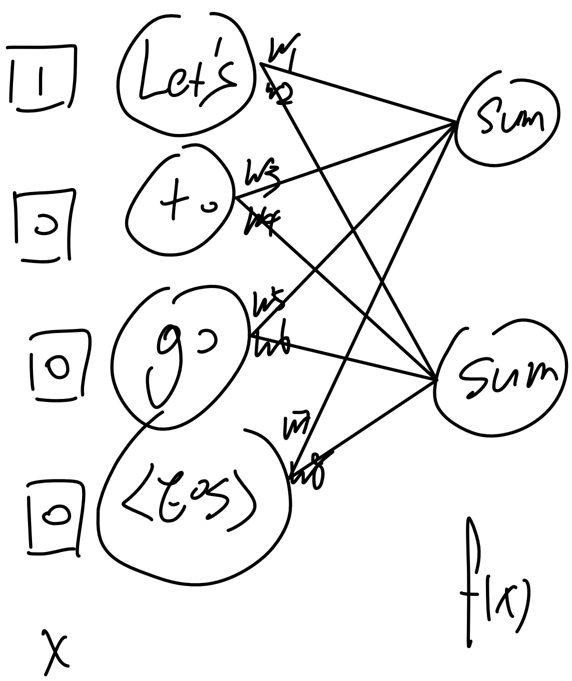
⇓

☆ Text Detection (NLP)

Transformer

① Word Embedding

$\boxed{\text{Text} \rightarrow \text{Number}}$



→ $1 \times 1.87 = 1.87 = 1.87$
$+$
0

→ $1 \times 0.09 = 0.09 = 0.09$
$+$
0
⋮

determine?

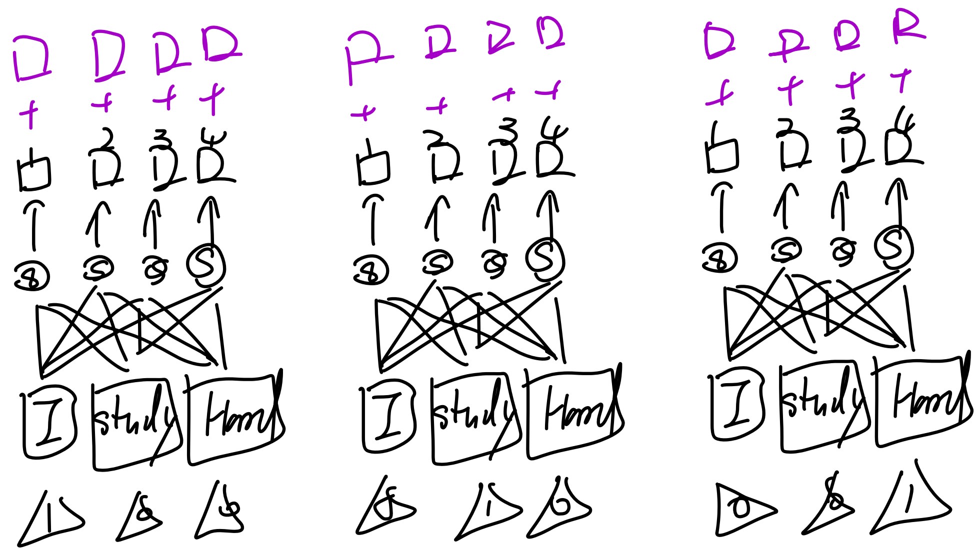
Let's $= (1.87, 0.09)$

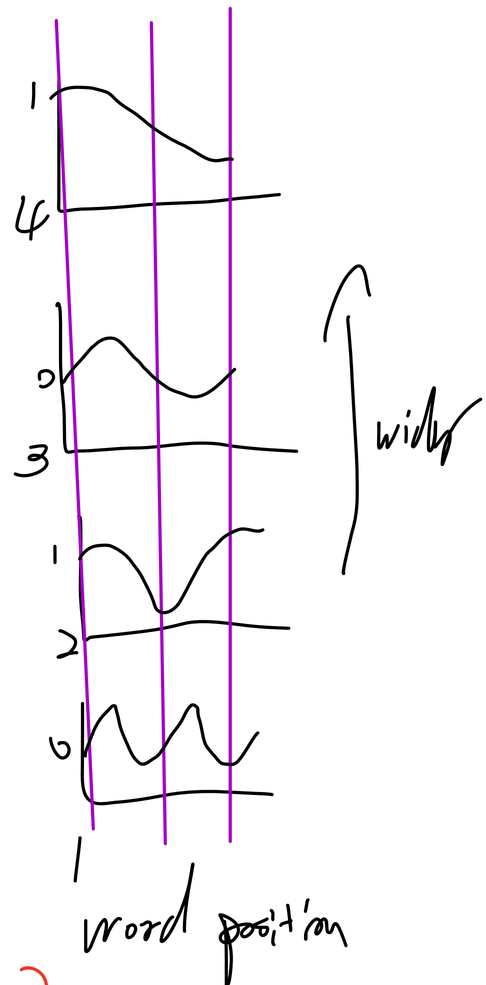
↑ same weights

$g_o = (-0.78, 0.27)$

weights determined by Backpropagation

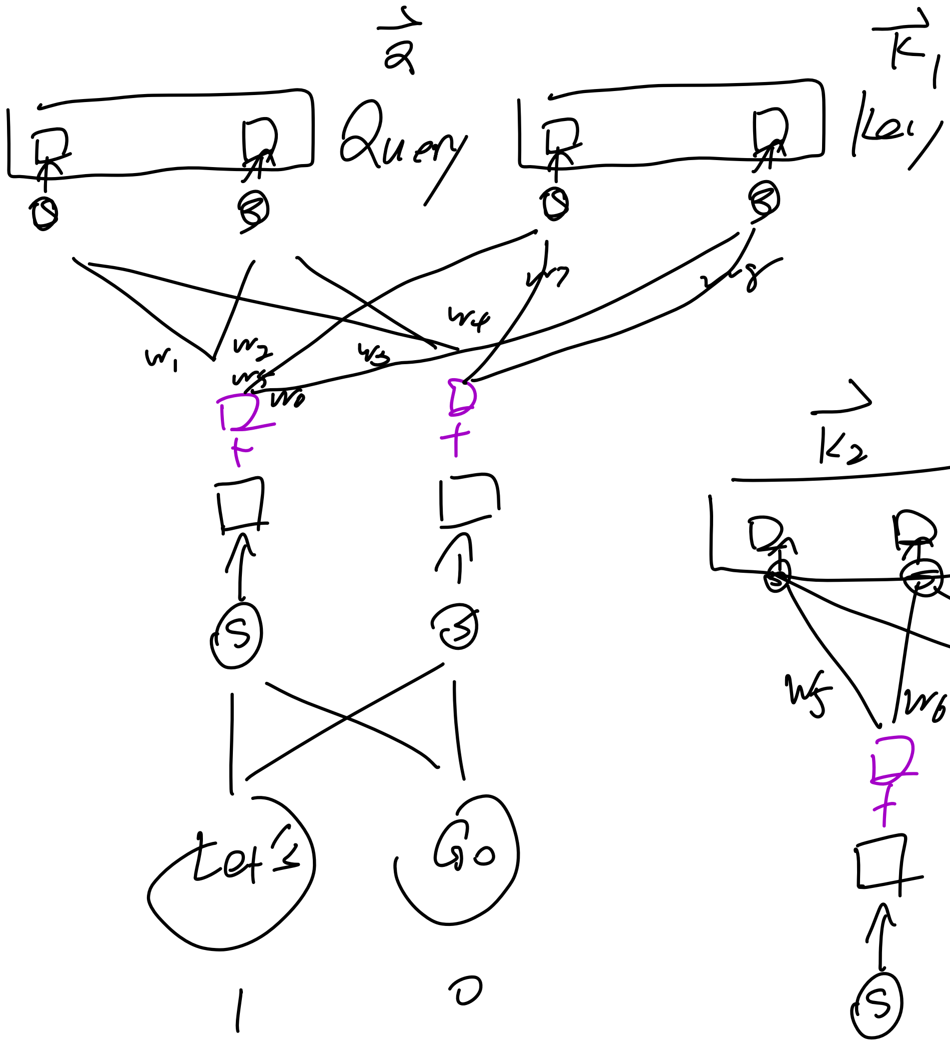
② Positional Encoding

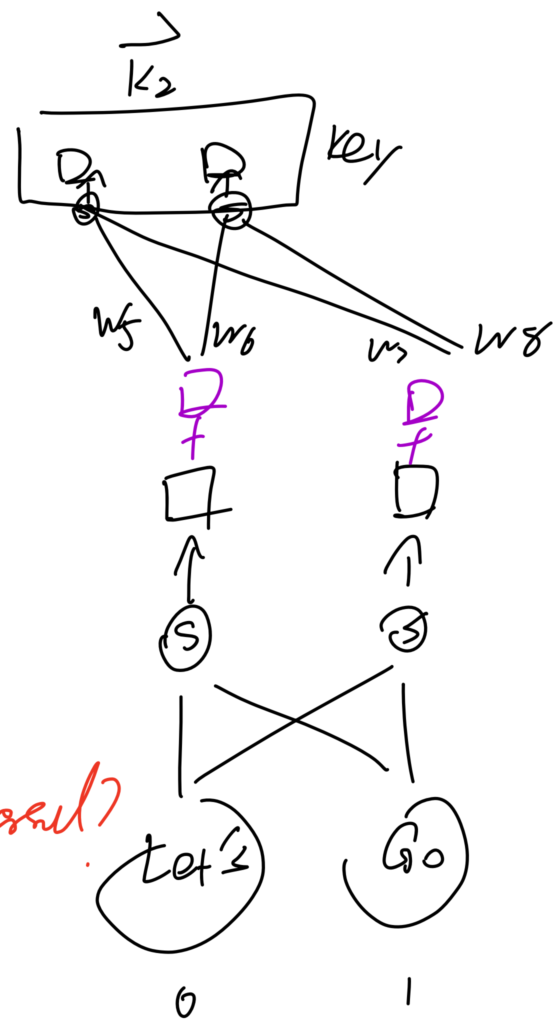




⇒ unique as word embedd more
+ wider

③ Self-Attention



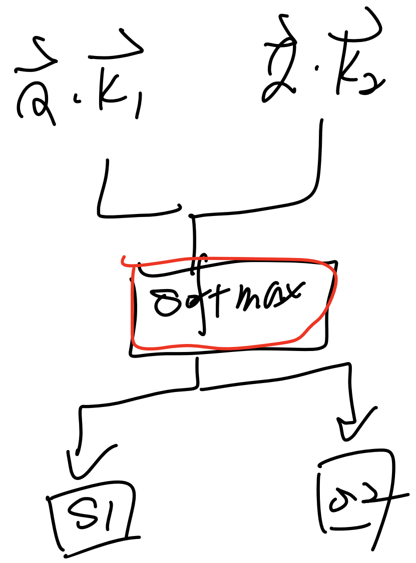


How but not Reversed?

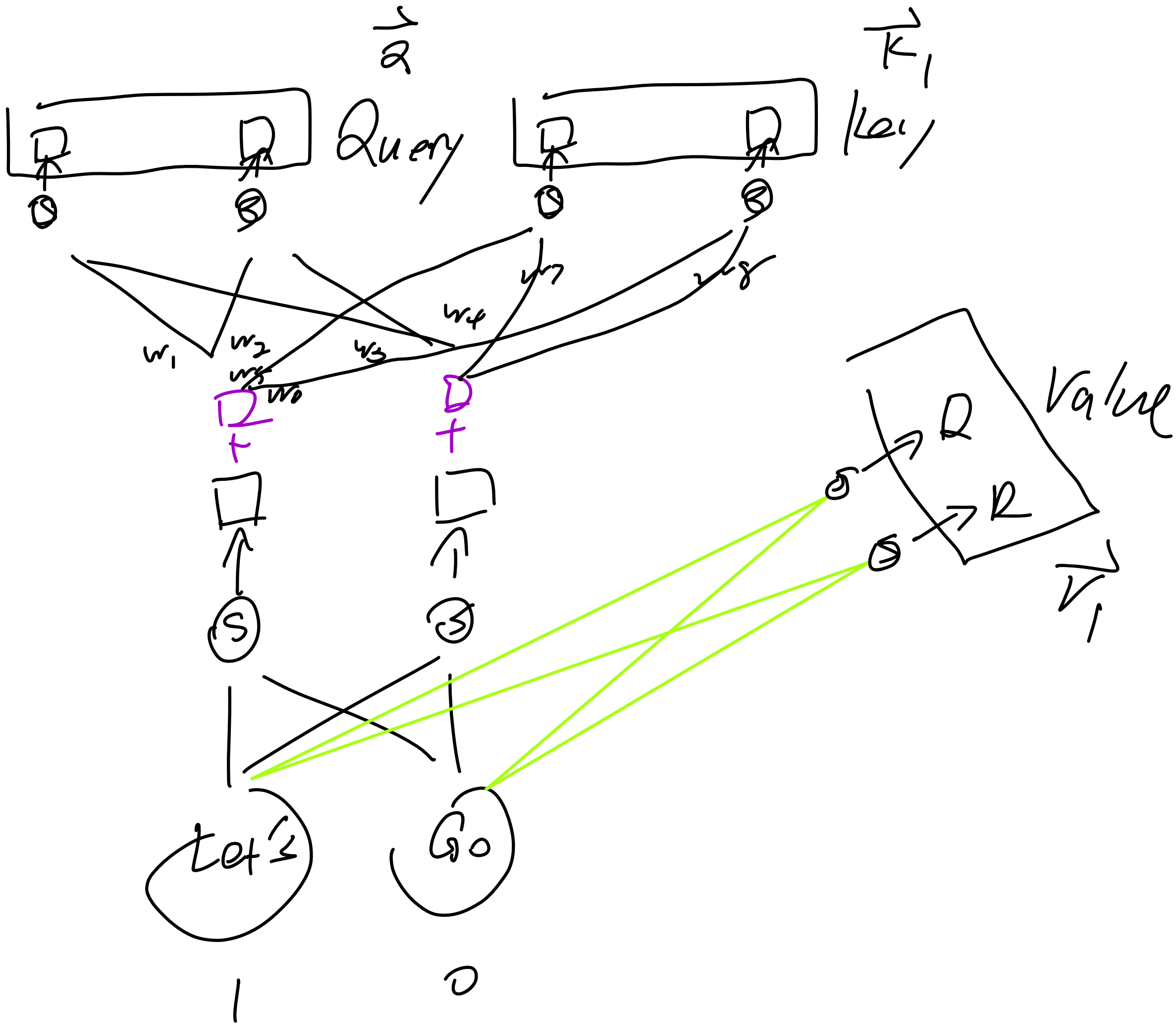
$\vec{q} \cdot \vec{k}_1 = 11.7$ $\qquad$ $\vec{q} \cdot \vec{k}_2 = -2.6$

$\Rightarrow$ $\boxed{\text{Let's go Let's}}$

I sell the stock since it shows bad performance



Q · K1
Q · K2
softmax
S1
S2



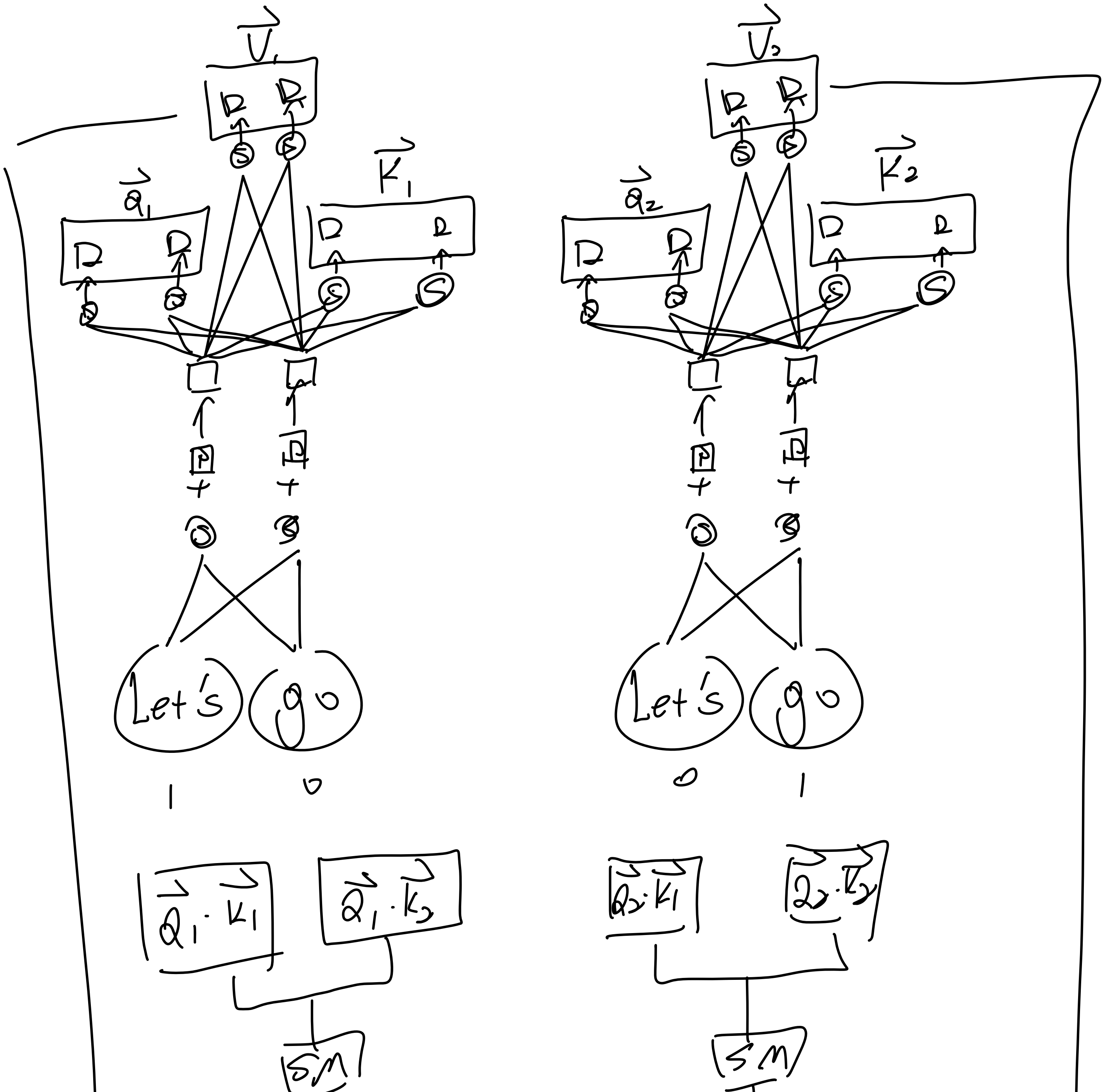
Q
K1
Query
Key
w1
w2
w3
w4
w5
w6
w7
w8
Value
V1
Let's
Go
1
0

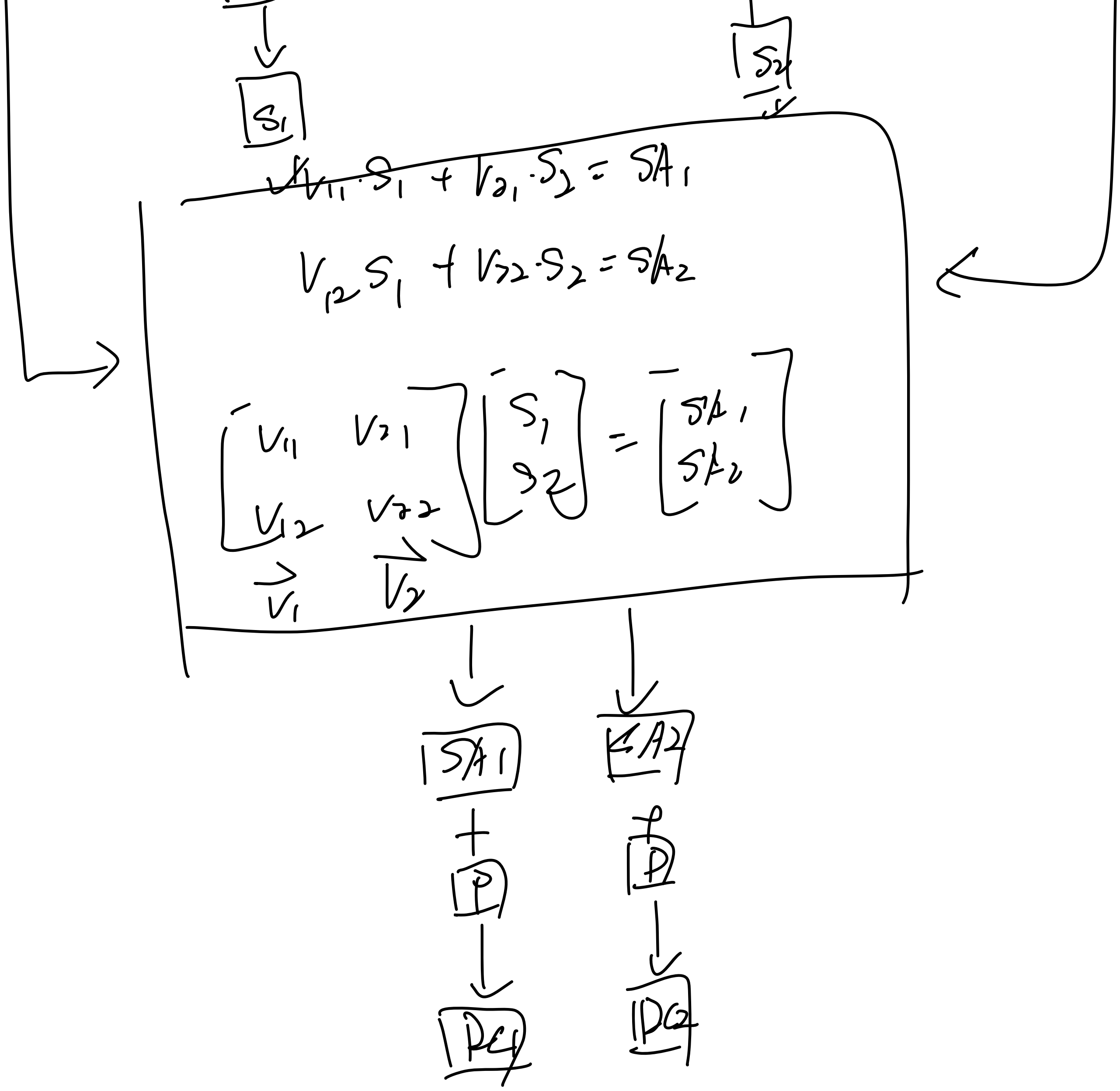
$$\vec{V}_{11} \cdot S1 \qquad \vec{V}_{12} \cdot S_1$$

$$+ \vec{V}_{21} \cdot S_2 \qquad \vec{V}_{22} \cdot S_2$$

$$= S_1^* \qquad = S_2^*$$

= self-attension value for Let's

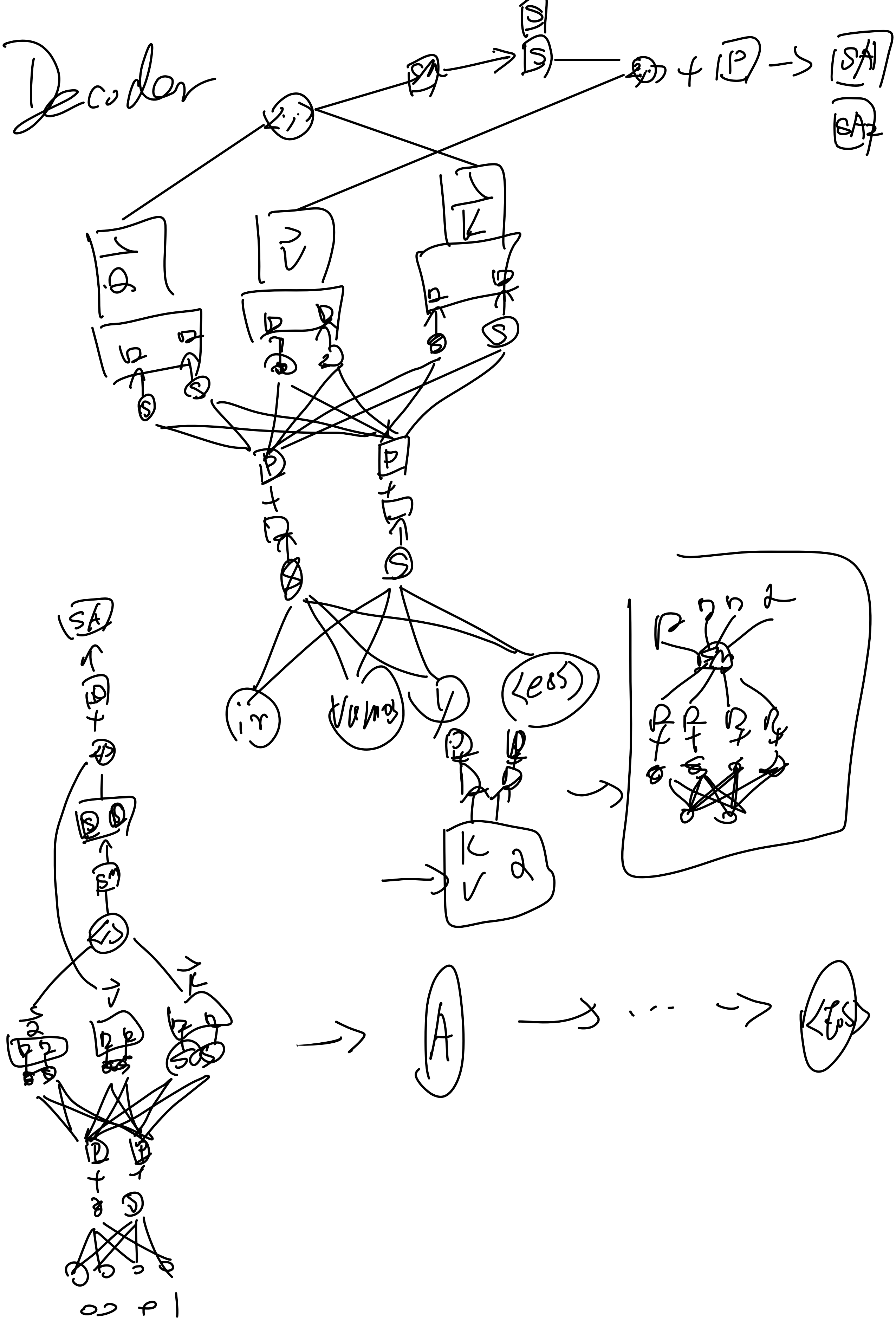




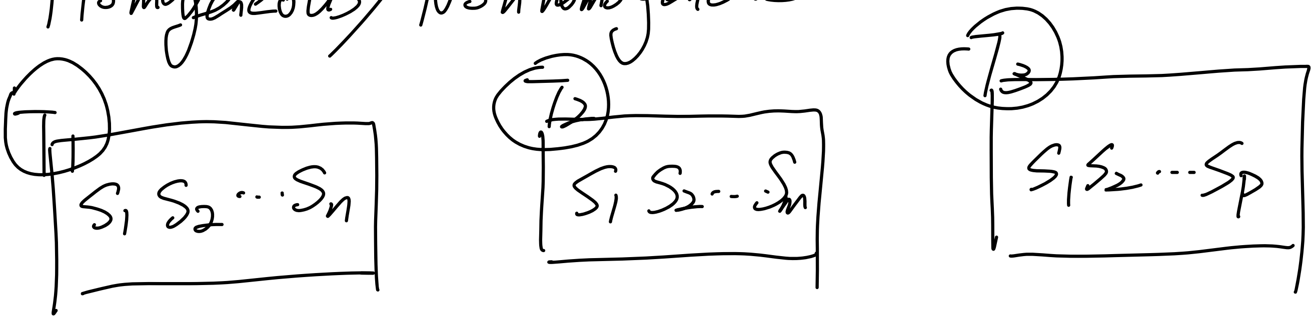
Word Embedding → Positional Encoding → Self-Attention → Residual Connection

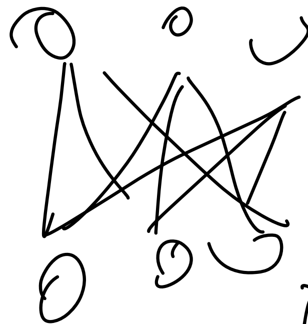
Encoder

Decoder

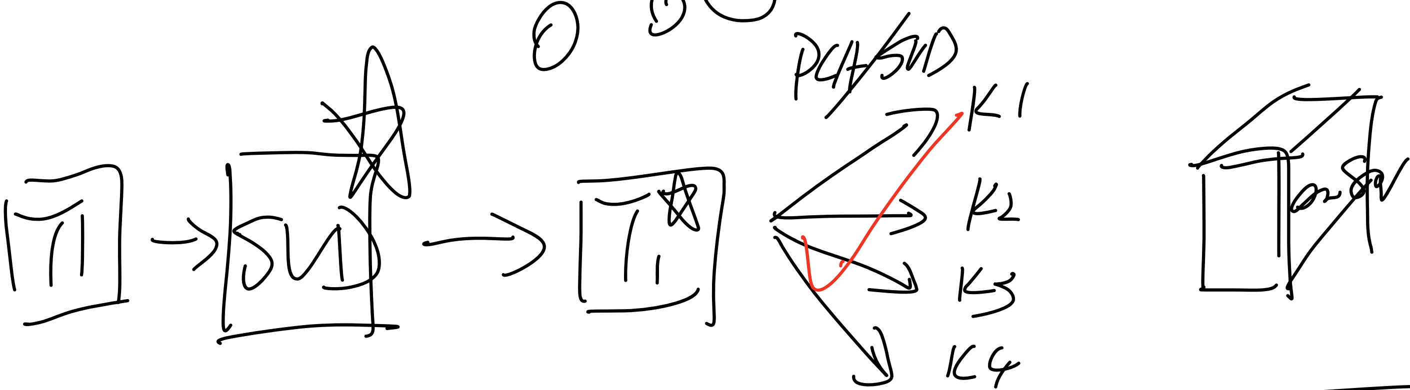


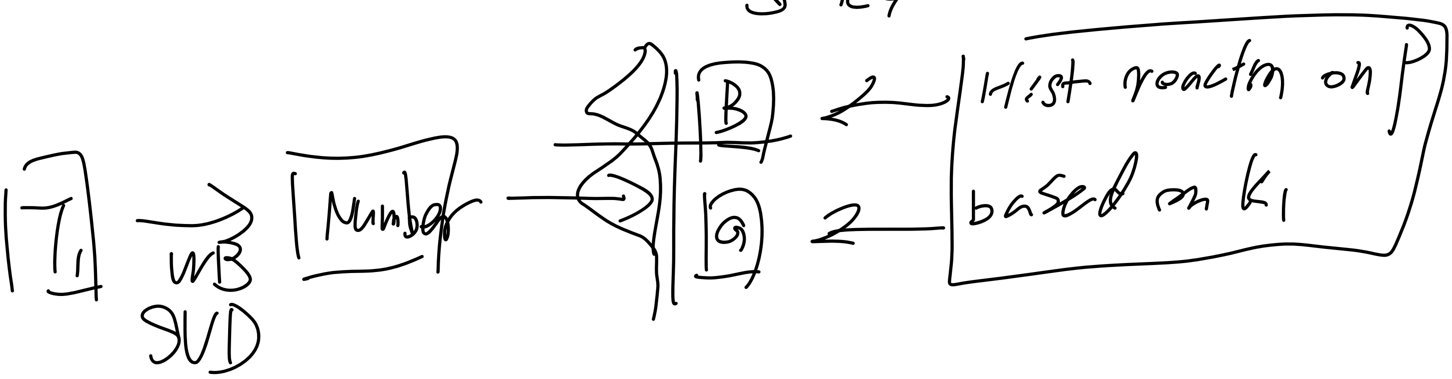
Homogeneous/Nonhomogeneous





one-hot word2vec





to make sure stochastically uniform in t

$P(t, k_1, k_2)$

why Res min?

Homogeneous / Nonhomogeneous

Define $T: S \longrightarrow V$

S is sentense space

V is normal vector space (Euclidian Norm)

Then $T(s_1) = v_1 \quad \cdots \quad T(s_n) = v_n$

T is the same as Word Embeding

so $T(\cdot) \Leftrightarrow WB(\cdot)$

Similarly,

we have $P(\cdot)$, $SA(\cdot)$ as positional Encoding and Self-Attention

$$\Rightarrow \quad TEncoder(\cdot) = SA \circ P \circ T^n$$

similarity among words

order infor

WB

Note $\boxed{\text{Word2vec}}$ can be referenced.

Define $\boxed{\text{Coordinator } C}$

$$\vec{K_t} = svd(TEncoder(T))$$

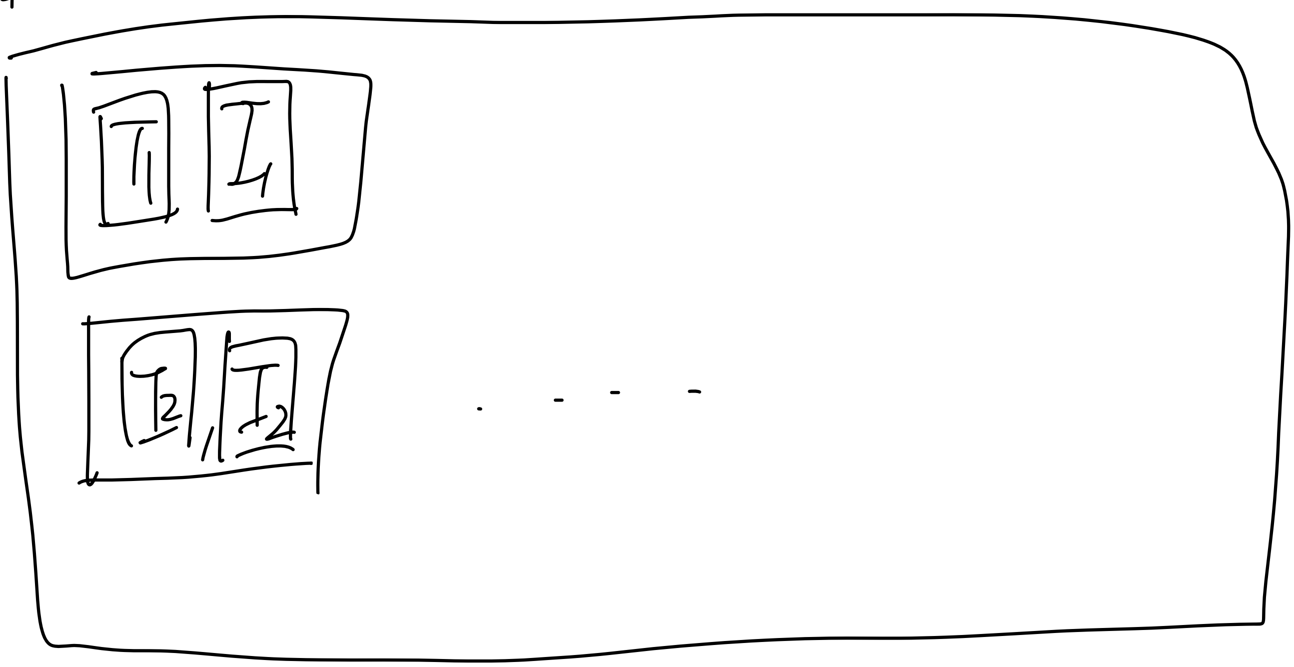
$$C(\vec{K_t}) = \Phi(t)$$

$$\Rightarrow P(t, \Phi(t))$$

Influence Prob

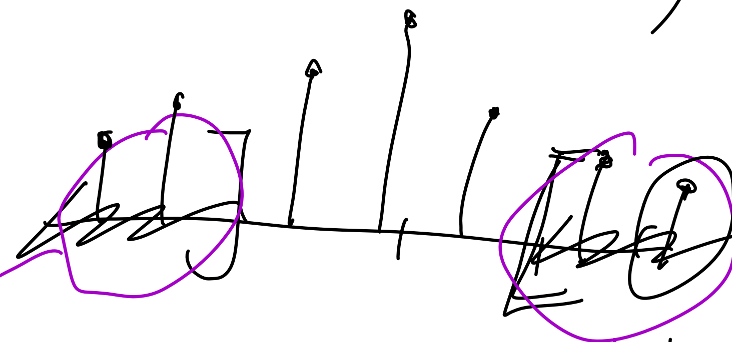
Determine a set of data with cons. to learn weights to train loss function

Data =



$\boxplus = \text{TEncoder}(\text{Text})$

$$(I_1, I_2, \dots) \xrightarrow{\text{Normalized}} (i_1, i_2, i_3, \dots)$$

$\boxplus \longleftrightarrow$ 

$$\text{svd}(\boxplus) = (k_1, k_2, k_3, \dots)$$

$$\max \ \text{svd}(\boxplus)$$
$$\text{s.t. } (i_1, i_2, i_3, \dots)$$

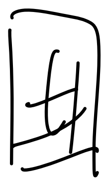
$$\min \ \text{svd}(\boxplus)$$
$$\text{s.t. } (i_1', i_2', \dots)$$

$\max \text{svd}(T_N)$
s.t. i_N

?

$\{w_{N,i}\}_i$ $\quad$ $\max\text{svd}(T_{N-1})$
s.t. i_{N-1}

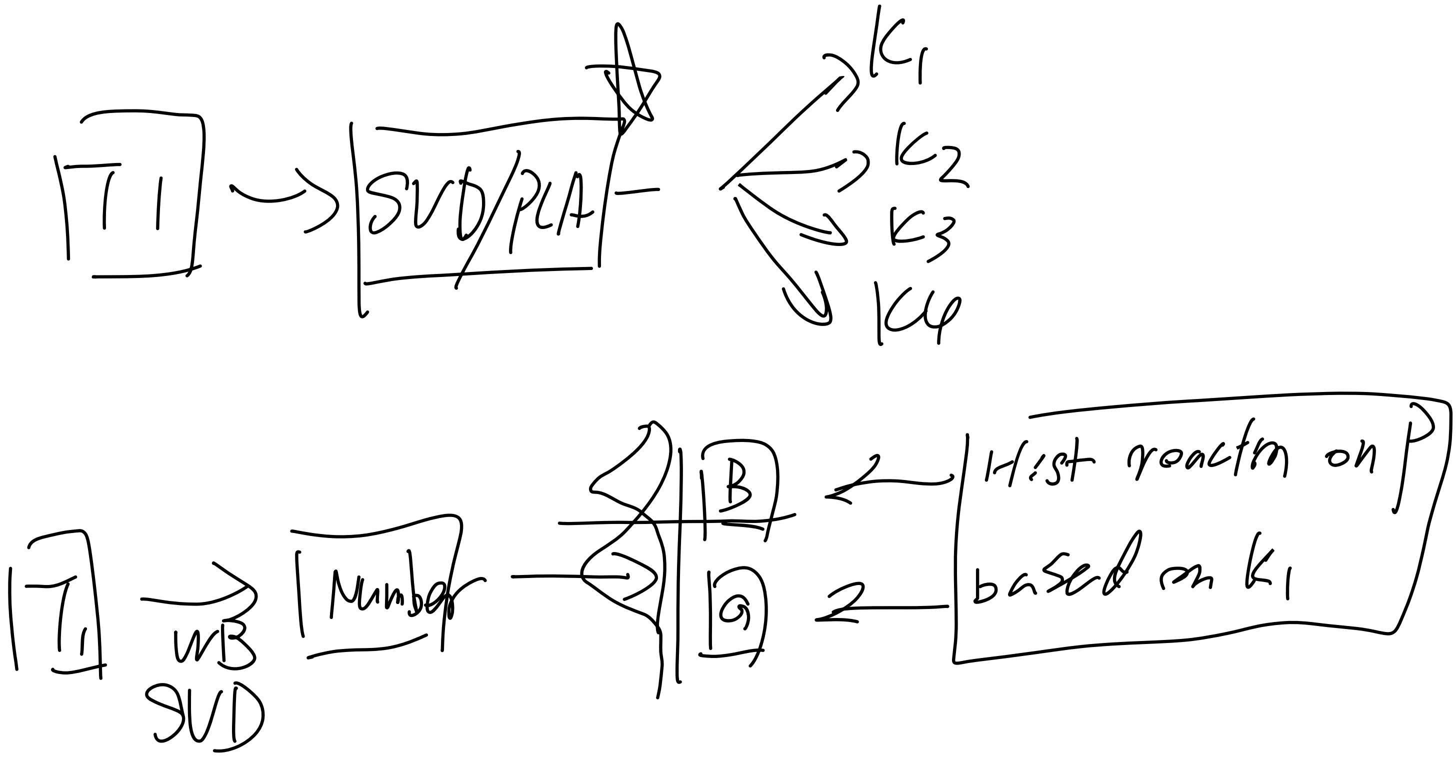
$\{W_{N-1,i}\}_i$

$\text{TEncode}(T) =$ 

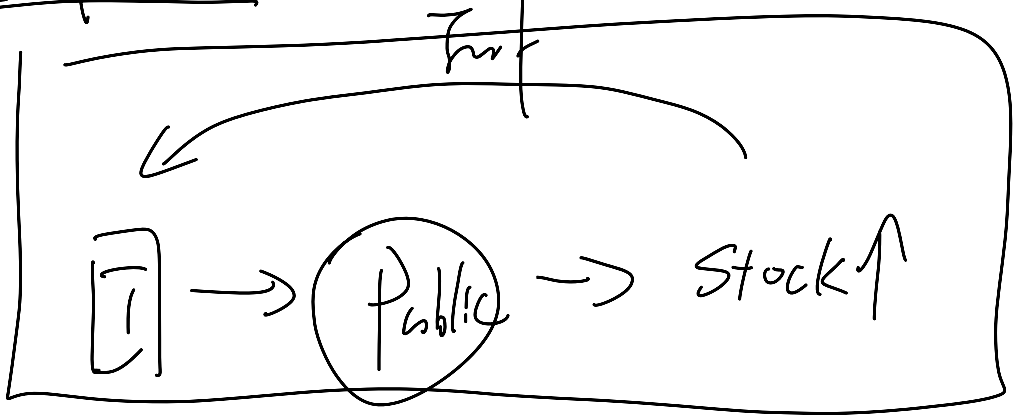
$\vec{W}_N \quad \vec{W}_{N-1} \quad \vec{W}_{N-2} \quad \cdots \quad \vec{W}_1$

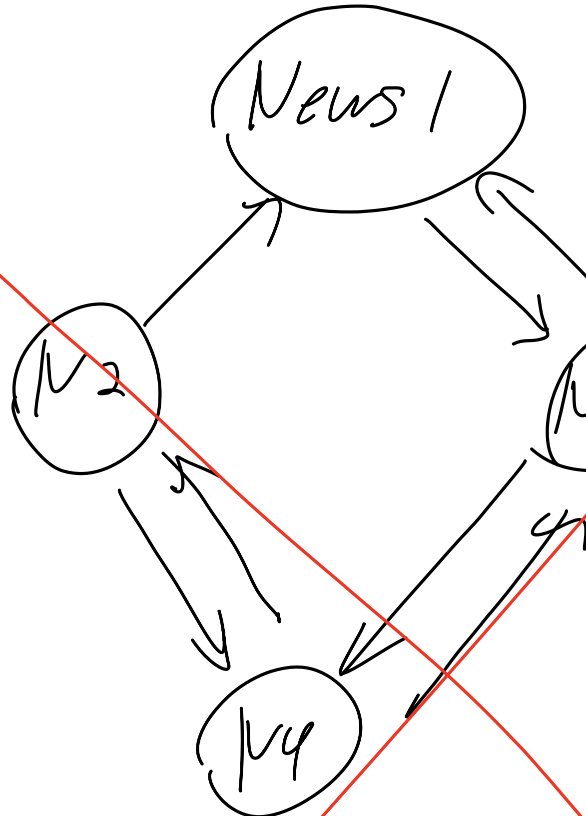
$$\begin{bmatrix} W_{N,1} \\ \vdots \\ \vdots \end{bmatrix} \leftrightarrow \begin{bmatrix} \ \end{bmatrix} \leftrightarrow \begin{bmatrix} \ \end{bmatrix} \leftrightarrow \begin{bmatrix} \ \end{bmatrix}$$

Interpolation


T1
SVD/PCA
K1
K2
K3
K4
T1
WB
SVD
Number
B
G
Hist reaction on P
based on K1

Influence





$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \Rightarrow \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} & & & \\ & \text{Matrix} & & \\ & & & \end{bmatrix} \end{array} \Rightarrow \quad \text{Eigenval } \lambda = 1 \quad v = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

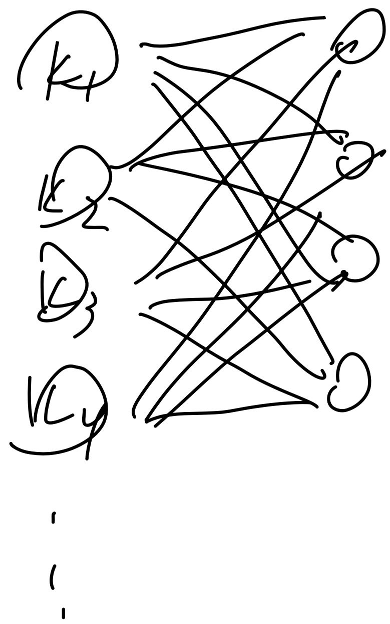
$$r_1 = \frac{q_1}{\sum_i q_i}$$

Possible Factor : Fast, Huge

or use ML to see more general factors for agents

Like NN

random data



Goal : Effectiveness

$\boxed{T} \longleftrightarrow \boxed{PC}$