# Attention and Transformers

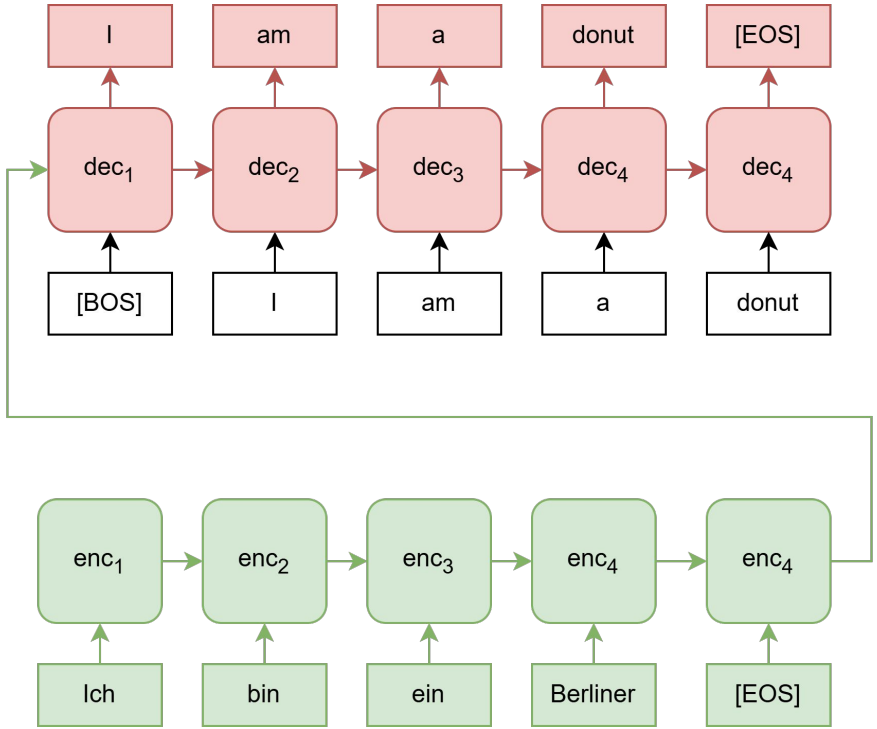
Lukas Edman

# Layout

1. RNNs recap
2. RNNs with attention
3. Transformer

# RNNs Recap

- Each token gets fed into the RNN one-by-one
- Hidden state is accumulated
- BOS token (or SEP token, whatever) initiates the decoding
- Given the hidden state and current token, output probabilities for the next token
- Take the argmax (if you're doing greedy search), or topk (if you're doing beam search)
- Repeat until reaching EOS



# RNNs Recap

Training

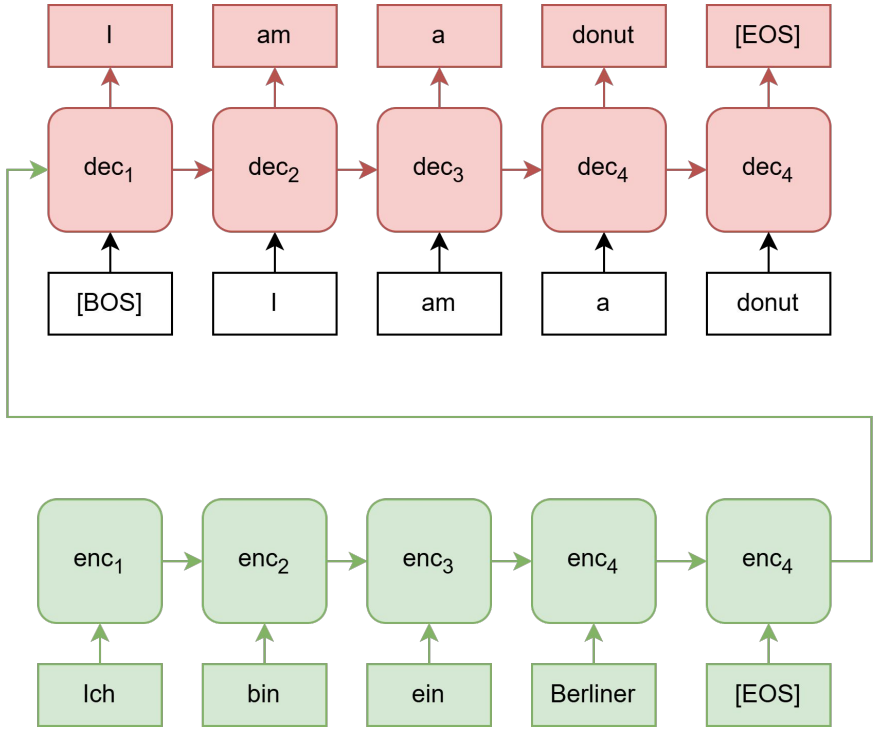
- Get output
- Apply softmax
- Cross Entropy Loss:

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i$$

where

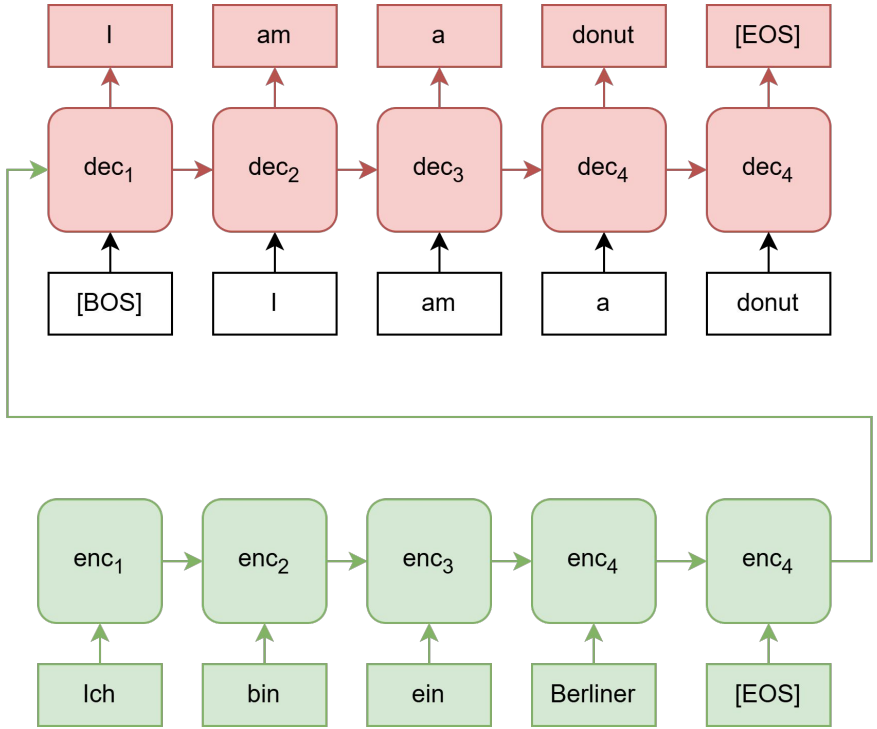
  - $\hat{y}_i$ = probability of token i
  - $y_i$ = whether $y_i$ is the correct token (1=yes, 0=no)
  - output size = vocabulary size

- Use teacher-forcing, e.g. if the model outputs "We" instead of "I", still train it as though it output "I"



# Problems with Vanilla RNNs

- What's the problem with this structure?
- Why might this structure make it difficult for an RNN to perform well?



# Problems with Vanilla RNNs

- It works well with short passages, but try feeding in:

Heute war ich in Zug (DB).

Es ist da was passiert, Geschichten die das Leben so schreibt. Sowas kann man sich nicht ausdenkeng.

1 alter Oma fuhr mit mir in ICE Zug DB von München nach Dresden, weiss selber nicht wie man darauf kommt durchzufahren, da ich Nürnberg raus wollte von noch in Bayern bleiben her. (Ich überkwere die Freiststaatsgrenzen nur selten, bin auch nicht geimpft etc). Vielleicht wollte sie dort Verwandte besuchen und bei Flucht helfen in Westen? KP aber es geht mir auch nicht an. Wie wir da so fuhren, kommt 1 Schaffner im Sinne von Kontrolle. Jeder packt Smartphone aus, 1 connected Laptop mit Schaffnerkontrollgerät von Netzwerk her ich zeige mein Smartwatch (Appel) für QR Code zum scannen. Omer sieht man schon an dass sie normales Papierfahrkarte hat, alle in Abteil sind schon am sie auslolen weil so rückschrittlich, passt einfach nicht mehr in die Zeit denken die wahrscheinlich. Ja und bissl recht haben ja auch.

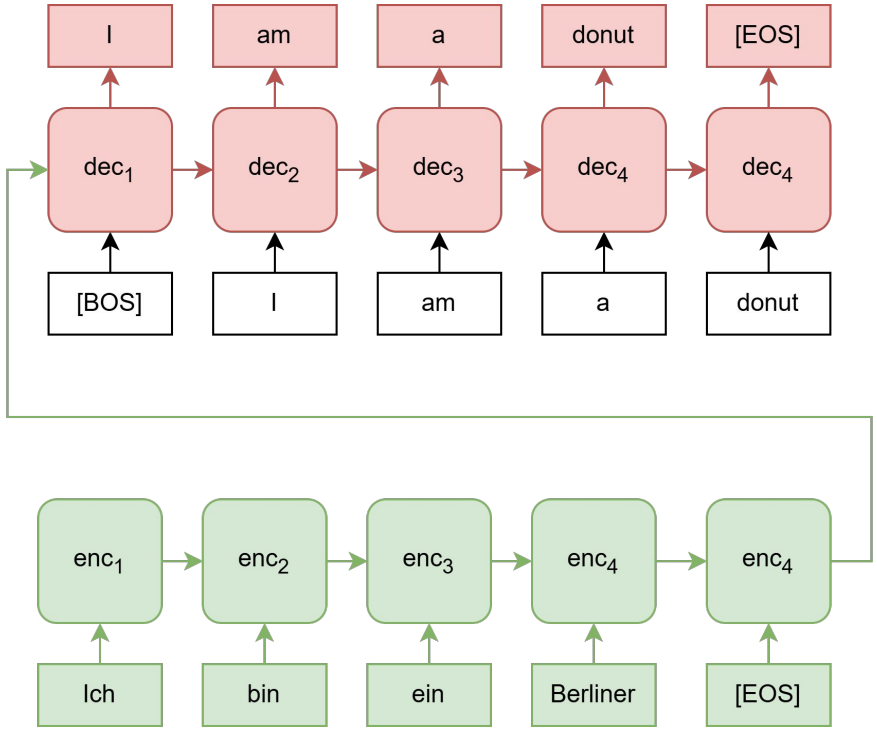
Aber langer Rede gar kein Sinn, es stellt sich raus die Omer hat auch noch 1 Ticket sich gekauft dass nicht im ICE Zug giltet. Dies hätte sie im Internet nachlesen können, aber das hat sie nicht weil es ist zu teuer und den VHS Kurs wo es erklert wird hat nicht leisten können. Da sind die Enkel gefragt meiner Meinung nach aber hilft jetzt auch nix mehr. Der Schaffner besteht auf seim Geld, das die alte Frau zahlen muss in Sinne von Schwarzfahren her es sind 120 Euro. Sie hätte nur fahren dürfen in Regionalbahn wo es 5 Tage dauert bis man Augsburg ist lol (von Münchem aus wohlgemerkt). Die Frau sagt hat kein Geld und sie muss vom Rente leben von ihrem verstirbten Mann und es ist wenig. Aber der Jockel von Schaffner lässt sie nicht erweichen. Er bleibt hart. Alle schauen verlegen auf ihrem Handy oder lesen Börsenkürse. Ich stehe auf und erhebe Wort. "Hör mal zu du Überjochen, die Frau fährt ja wohl jetzt unsonst, sonst haben wir hier rukizuki Rambazamber und es gibt 1 Bombe" sage ich den Schnauzbartträger im Gesicht. Es wird noch leiser in Abteil, was gar nicht möglich ist weil ja vorher schon so leise war. Er sagt kann er nicht machen wegen Privatisierung von Bahn früher schon da hätte es Steuerzahler gezahlt aber jetzt nicht und ihm sind auch die Finger gebunden. Ich erwiedere: "Schau mal her du Lauchkönig, ich zahl jetzt die Hälfte von dem Schwarzfahren aber dann ist auch gut, den Rest zahlt die DB der Knechtzirkus" Jetzt stehen Leute auf und wollen Schlegerei anfangen von sozialer Ungerechtigkeit her. Sie sind entzürnt weil die Bahn so 1 Geldverein ist. Einer rollt das DB Kundenmagazin gans fest zusammen, für dass er ordentlich zubatschen kann. Der Schaffner ist in der Unterzahl (logisch). Allgemeine Berlinstimmung jetzt in Abteil, Kaffeebecher und Kundenmagazine fliegen durch die Luft (Rigaerstrasse artig)

Die Omer ist das peinlich, sie möchte das nicht das soviel Trurbel um sie gemacht wird. Ich sage "Fresse jetzt Hexengesicht, es ist 1 Sache von Ehre jetzt. Dem Opferkönig sein Zahnbürschte greift morgen ins Leere wenn es jetzt weitergeht und ich aufdrehe von Fausttanz her" Gegröhle in Wagen. 1 mitreissender Arzt hat sein Koffer aufgemacht und 1 Flasche Chlorerform hervorgeholt für dass Betäubung da wäre wenn man bräuchte in Kampfgetümmel.

Jetzt gibt der Schaffner klein bei. Er sagt es passt so und er würde Omer nicht mehr belestigen. Ich gebe ihm noch die 60€ und klatsche ihm auf die Stirn wie man das mit schlechter Schüler macht. Unter hemischen Rufen wie "Du Protojockel!" und "Hau ab du Vollgasotto!" verlesst er das Abteil. Die Omer dankt allem und sätzt sich verlegen hin. Es war viel für sie. Von Aufregung her.

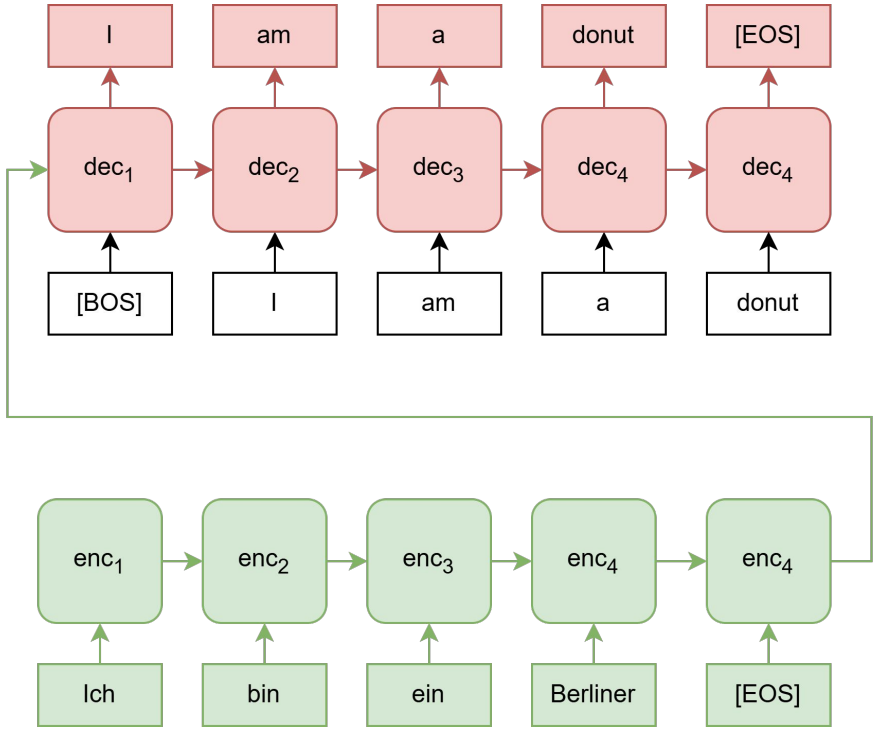
Die Leute wollen jetzt auch mir Geld zustecken weil sie Aktion gut fanden glaub ich. Ein modischer Gescheftsmann aus Bad Tölz steckt mir beileufig 1 Hunni zu und meint verschmitzt "Ich kann mir gönnen ich hab 1 Startup von veganer Käse es boomt" An Ende habe ich 460 Euro plus. Da sieht man dass es sich auszahlt wenn man Solidarität und Zivilcourtage zeigt. Alleine sind wir schwach, gemeinsam sind wir mehrer! #zivilcourtage #alleinesindwirschwach #ottohaftigkeitabschaffen

- Then it won't work so well.



# Problems with Vanilla RNNs

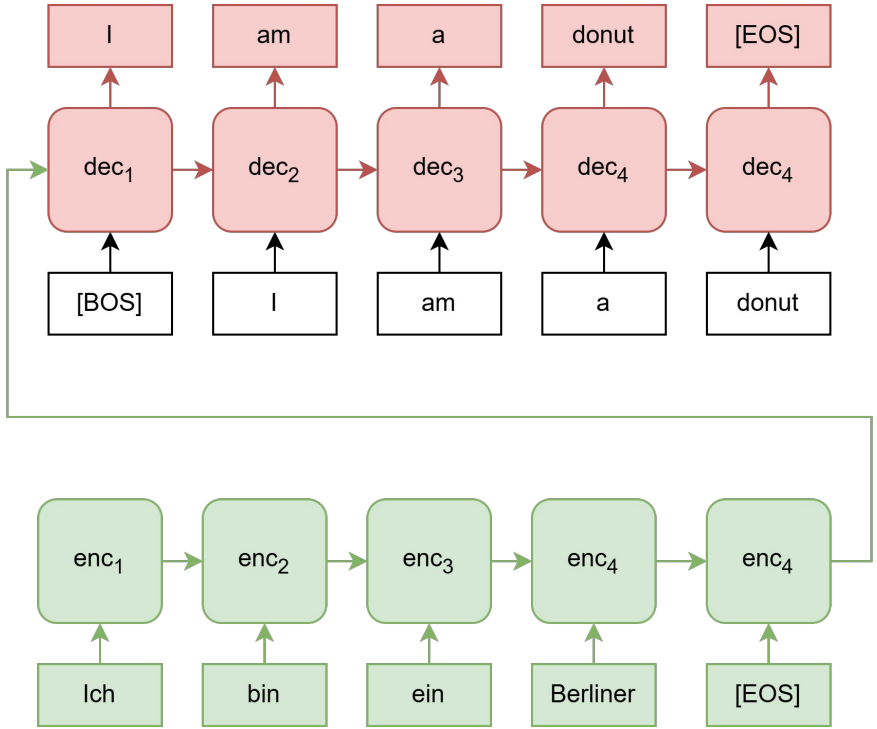
- There is a bottleneck
- The hidden state has to accumulate all of the input, so it becomes harder to accumulate everything perfectly (losslessly) the longer the text gets



# Problems with Vanilla RNNs

- There is a bottleneck
- The hidden state has to accumulate all of the input, so it becomes harder to accumulate everything perfectly (losslessly) the longer the text gets

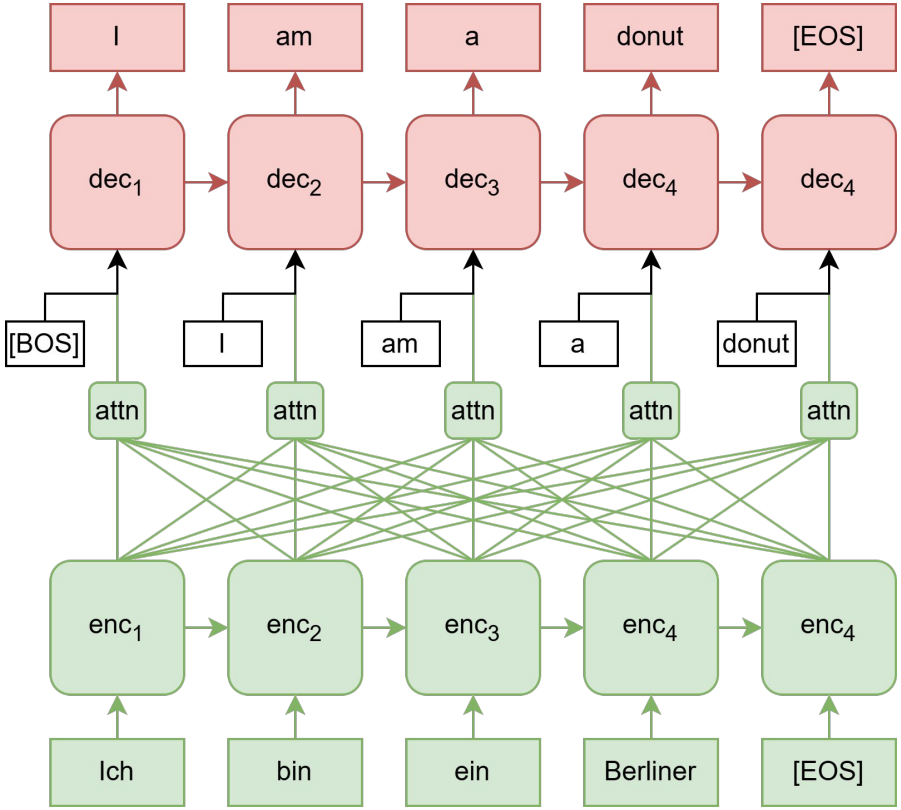
- So what's the solution?



# RNNs with Attention

- Attention lets the model see everything from the input altogether
- All of the input states get aggregated and are fed in to the RNN decoder at each step
- The additional connections remove the bottleneck

But what's the "attn" block doing?



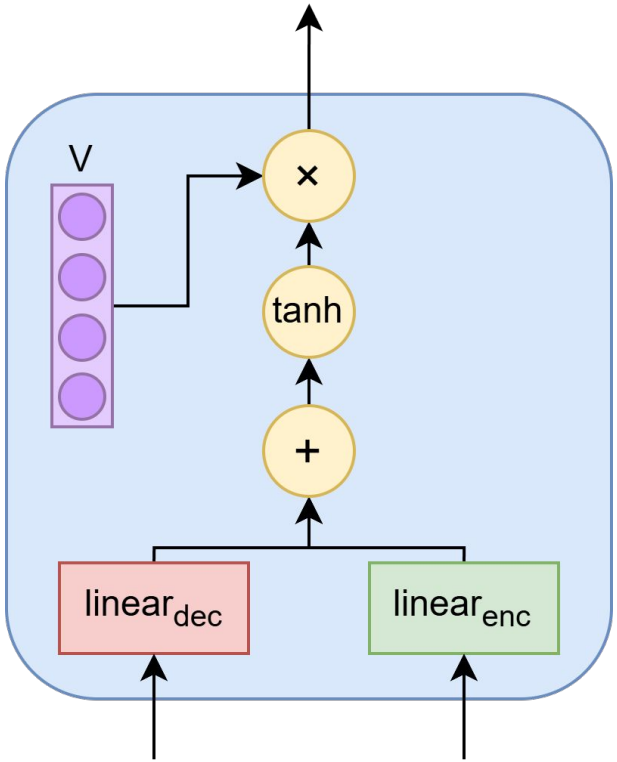
# Attention Block

- First we calculate an attention score $e_{t,i}$ for each input *i*, at timestep *t*

$$e_{t,i} = \text{score}(h_t^{dec}, h_i^{enc})$$

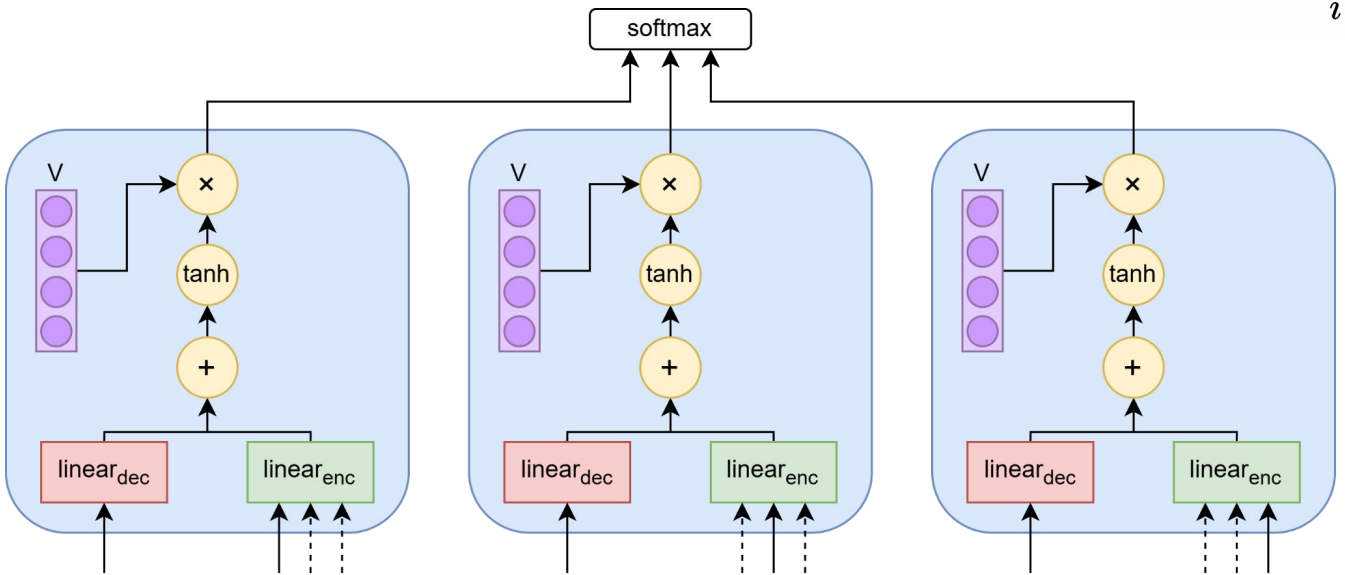
$$= v^T \tanh(W_1 h_t^{dec} + W_2 h_i^{enc})$$

- Repeat this for every token in the input (i in (0, N))



# Attention Block

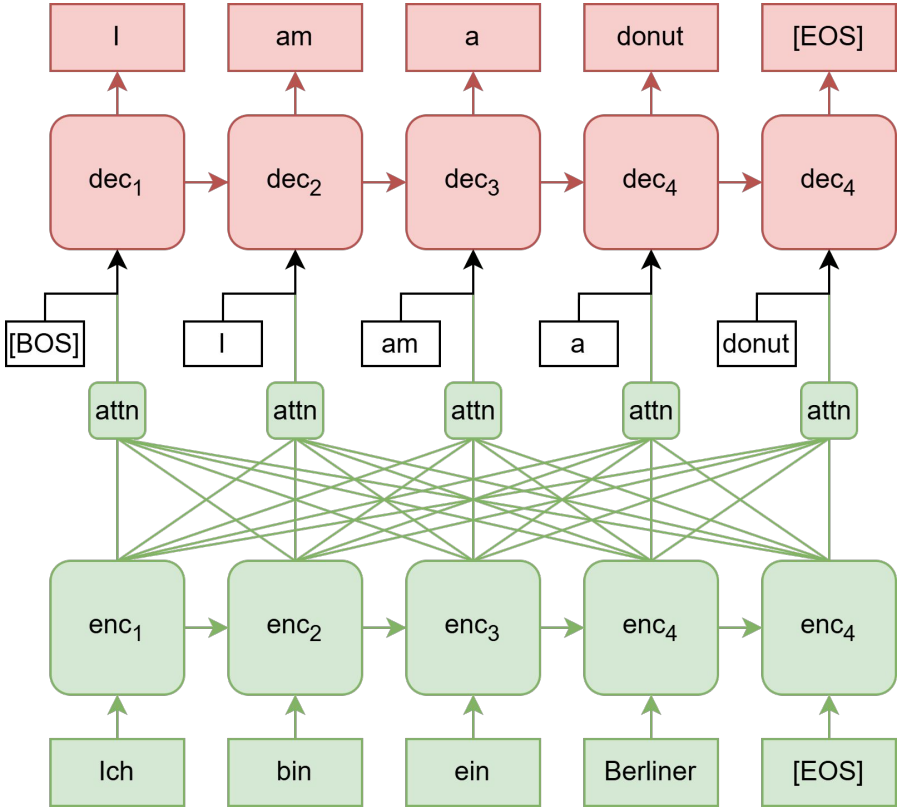
- Take the softmax of each encoding: $\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_j \exp(e_{t,j})}$
- Use the attention scores as weights for a weighted average: $c_t = \sum_i \alpha_{t,i} h_i^{enc}$



# RNNs with Attention

- RNNs with attention work a lot better, especially for longer sequences

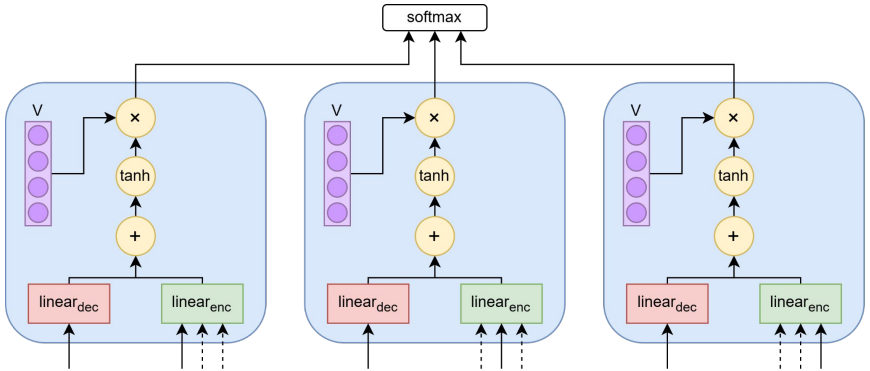
- But we can do even better…



# Transformers

# Is Attention All We Need?

- RNNs are pretty slow, given that they have to process each token sequentially
- Hold on, this attention thing just let us aggregate all of the input side in parallel steps…
- Why don't we make the whole architecture parallel and save massive amounts of time?




We can do these 3 all at the same time!

# Attention *IS* All We Need!

- Model architecture based on attention
- Multiple layers of attention for lots of contextualization
- Every token can be processed in parallel in training
- Generation must be done sequentially, but only for the target side

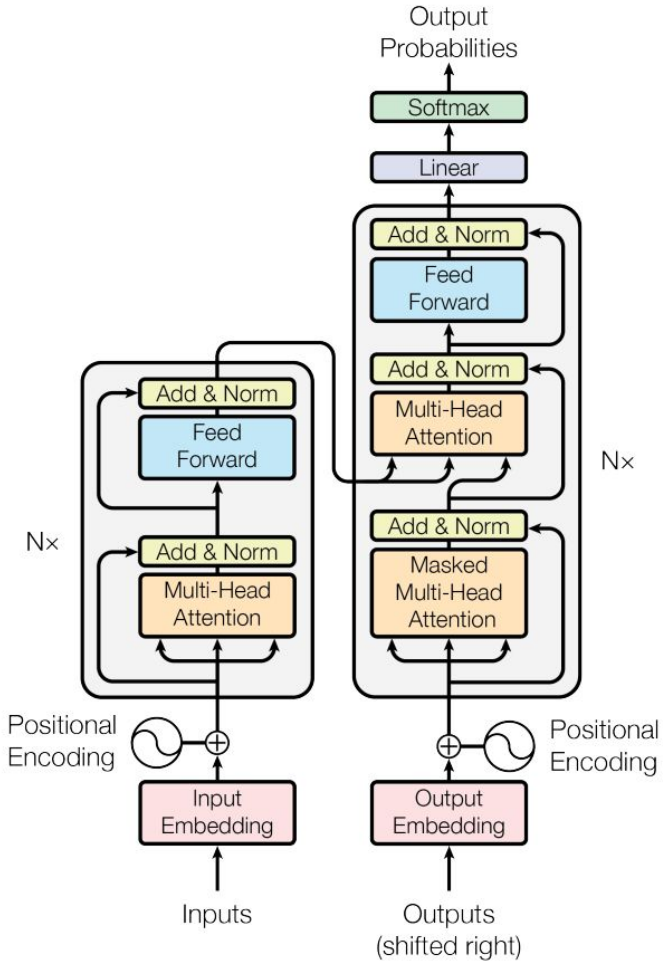



Figure 1: The Transformer - model architecture.

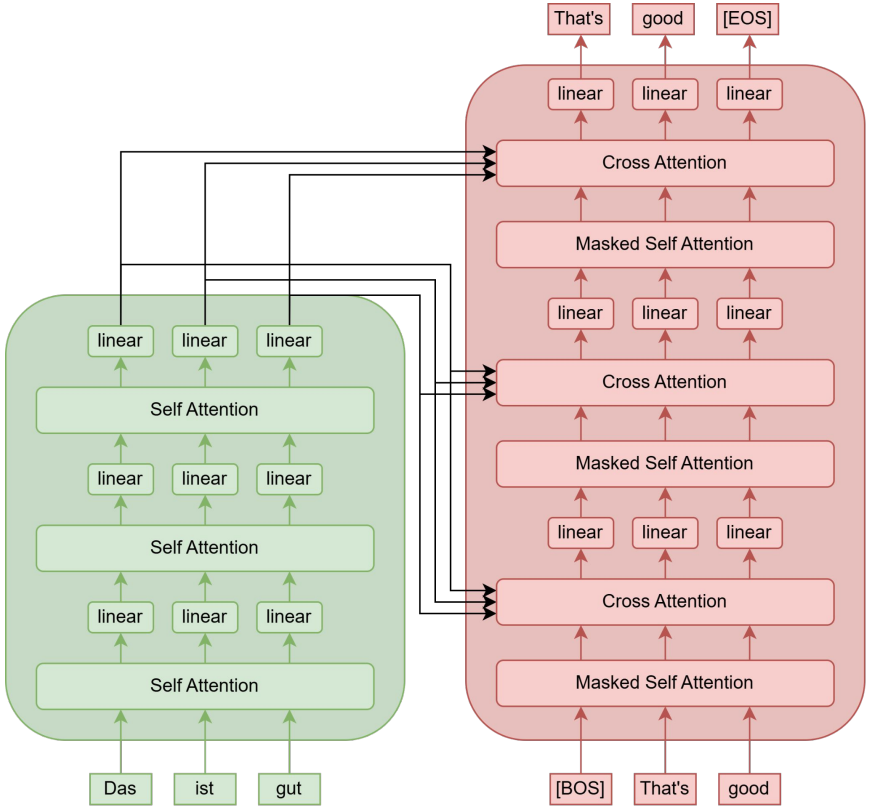
# Transformer Architecture

The architecture has 3 main components:

1. Self attention
2. Cross attention
3. Linear layers

The encoder is bi-directional (or more accurately, omni-directional)

The decoder is still uni-directional (left to right)



# Self attention

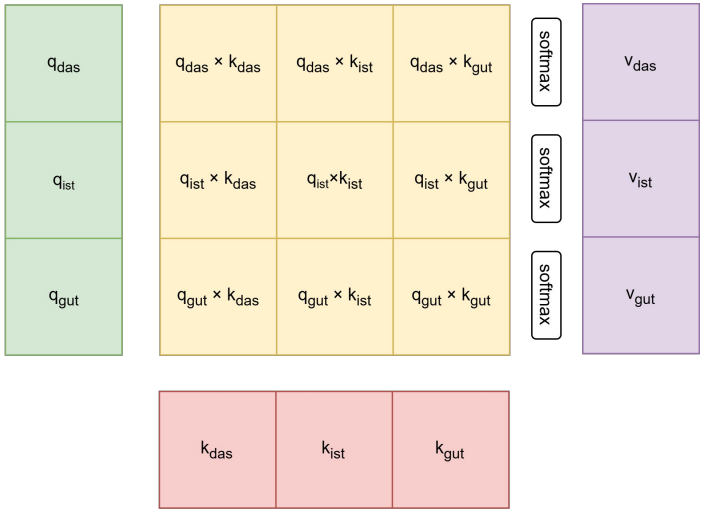
Very similar to the attention we saw earlier

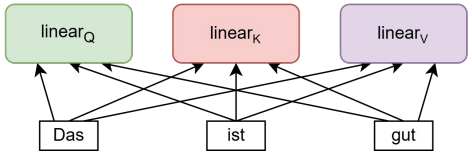
Get scores (e): $$e_i = \frac{q \cdot k_i}{\sqrt{d_k}}$$

Softmax: $$\alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)}$$

Multiply by value: $$c = \sum_i \alpha_i v_i$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



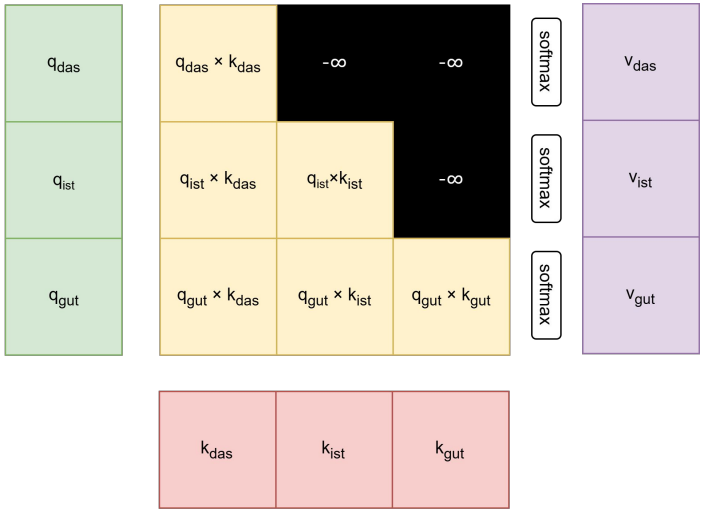



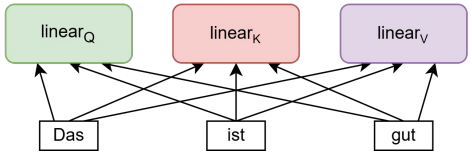
# Masked Self attention

In the decoder, we can't look ahead, that would be cheating!

Mask $QK^T$ such that $q_i k_j$ is $-\infty$

whenever j > i



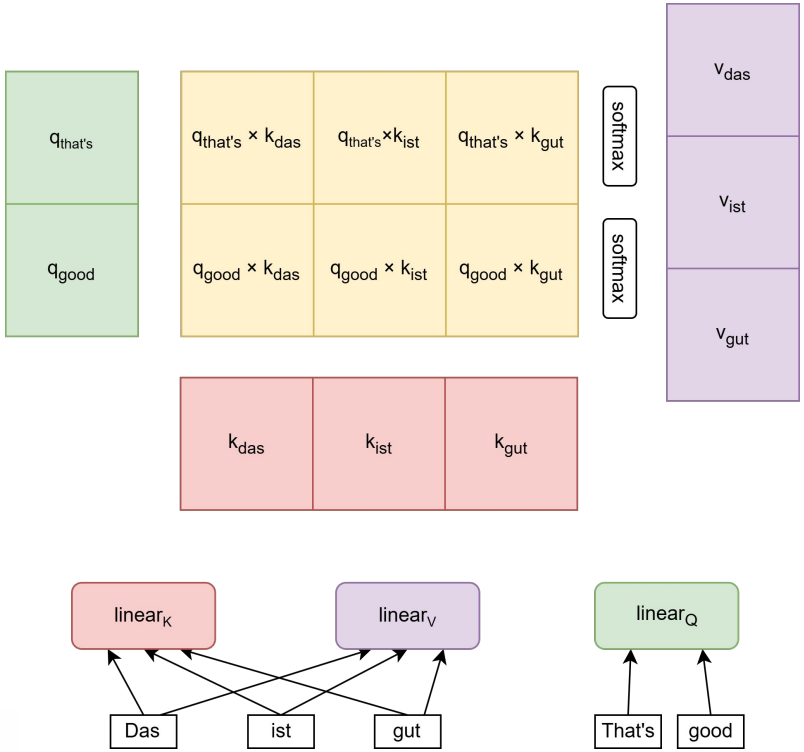

Because this is the only time tokens are mixed, the model cannot look ahead!

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

# Cross attention

Same as self attention between source and target, except that

- K and V come from the **source** side
- Q comes from the **target** side



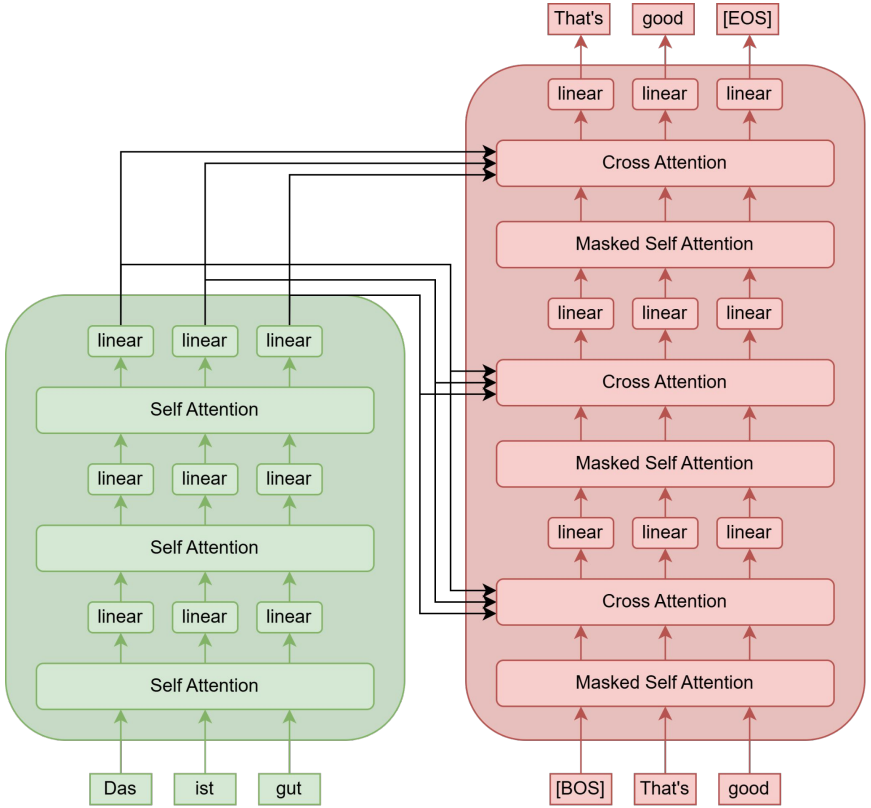

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

# Transformer Architecture

1. Self attention – Mixes tokens in encoder or decoder

2. Cross attention – Mixes tokens from encoder into the decoder

3. Linear layers – Transform the representation of each token *independently*



# Transformer Architecture

More minor things:

- Multi-head attention – Splits the hidden dim into smaller groups, computes attention on them independently
- Positional encoding – adds another embedding that encodes the position of each element.
- Layer Normalization – helps stabilize training.

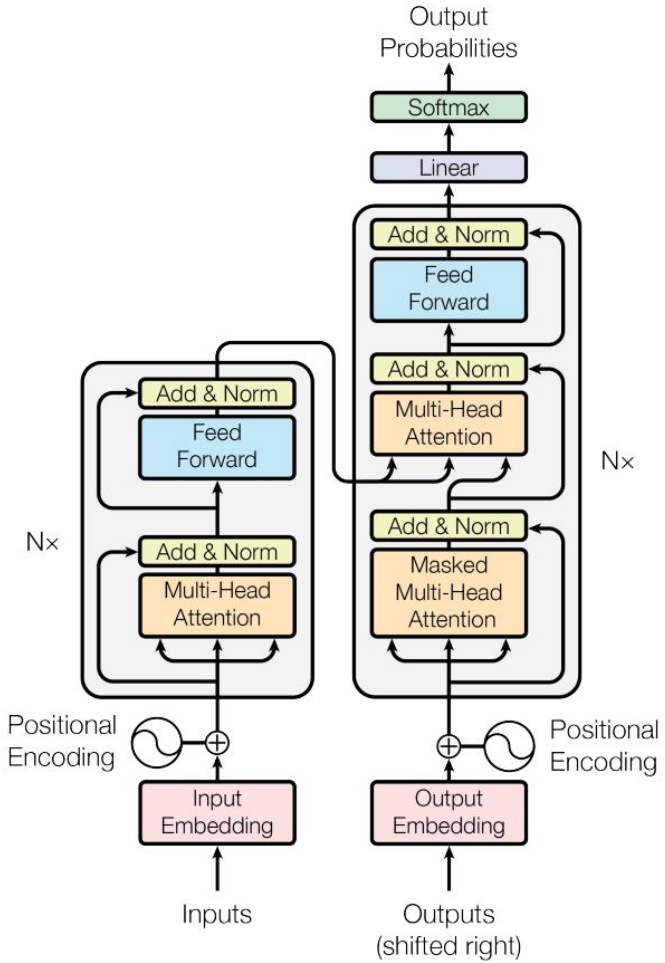
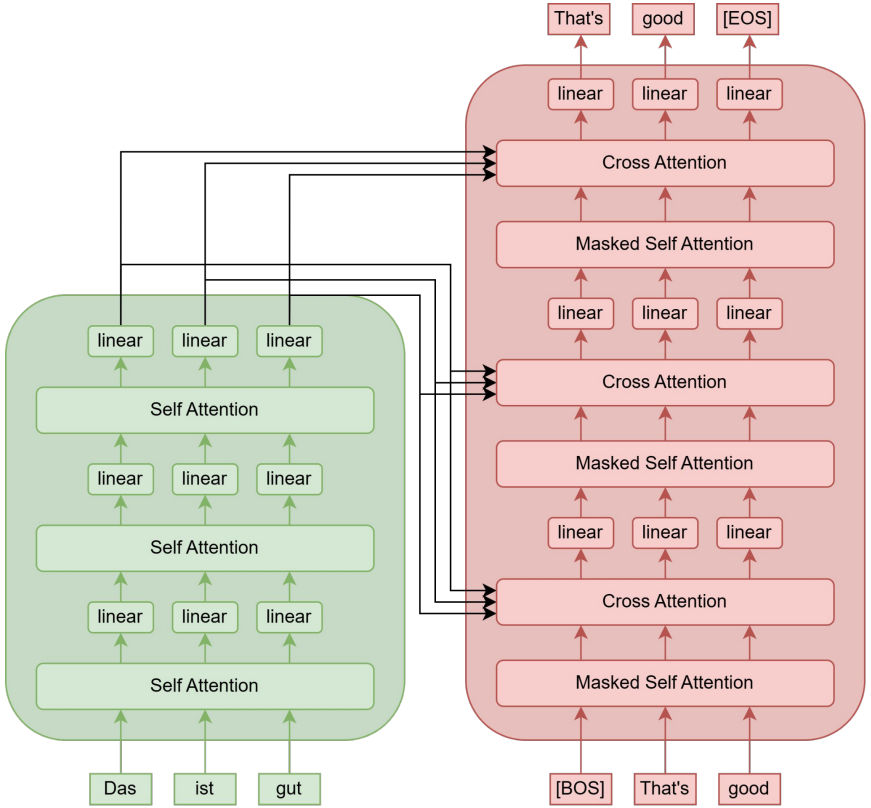



Figure 1: The Transformer - model architecture.

# Transformer Training

- Training in parallel → much faster
- General setup the same as RNNs:
  - Outputs probabilities
  - Uses cross entropy loss



# Attention in Transformers

You can probe attention in a Transformer and see what it's looking at

For a good translation model, you'll see something like this:


| | _Le | _développeur | _a | _argument | é | _avec | _le | _concepteur | _parce | _que | _son | _idée | _ne | _peut | _pas | _être | _mise | _en | _œuvre | . | </s> |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| _The | 0.958 | 0.713 | 0.127 | 0.434 | 0.218 | 0.029 | -0.091 | -0.296 | -0.063 | 0.165 | -0.018 | 0.157 | -0.045 | 0.388 | 0.044 | 0.016 | 0.225 | 0.037 | 0.248 | -0.092 | -0.086 |
| _developer | 0.202 | 0.694 | 0.376 | 0.434 | 0.047 | -0.105 | 0.089 | -0.171 | -0.101 | 0.125 | -0.143 | -0.163 | -0.168 | 0.087 | -0.084 | -0.018 | 0.047 | -0.086 | 0.054 | -0.111 | -0.009 |
| _argued | 0.104 | 0.068 | 0.842 | 0.781 | 0.548 | 0.068 | 0.059 | -0.009 | -0.053 | 0.242 | -0.053 | -0.062 | 0.003 | 0.161 | -0.041 | -0.078 | -0.071 | -0.058 | -0.082 | -0.079 | -0.036 |
| _with | 0.058 | 0.013 | 0.243 | 0.031 | 0.542 | 0.986 | 0.807 | 0.689 | 0.226 | 0.475 | 0.209 | 0.187 | 0.214 | 0.332 | 0.214 | 0.07 | 0.149 | 0.043 | 0.185 | 0.086 | 0.063 |
| _the | 0.044 | 0.051 | 0.083 | 0.064 | 0.299 | 0.078 | 0.314 | 0.595 | 0.914 | 0.47 | 0.289 | 0.21 | 0.348 | 0.189 | 0.215 | 0.133 | 0.193 | 0.116 | 0.145 | 0.097 | 0.127 |
| _designer | 0.029 | 0.044 | -0.049 | 0.004 | 0.331 | 0.002 | 0.356 | 0.212 | -0.022 | -0.025 | -0.072 | -0.082 | -0.025 | 0.208 | 0.019 | -0.072 | 0.108 | -0.068 | 0.058 | -0.053 | 0.022 |
| _because | 0.046 | 0.013 | 0.14 | 0.055 | 0.106 | 0.022 | 0.195 | 0.078 | 0.299 | 0.48 | 0.204 | 0.032 | 0.34 | 0.443 | 0.242 | -0.064 | 0.269 | 0.032 | 0.4 | -0.111 | 0.111 |
| _her | -0.007 | -0.008 | 0.118 | 0.003 | 0.07 | 0.017 | -0.245 | -0.017 | 0.074 | 0.424 | 0.863 | 0.316 | 0.096 | 0.203 | 0.121 | -0.067 | 0.11 | 0.018 | 0.255 | -0.022 | 0.055 |
| _idea | 0.007 | 0.016 | 0.079 | 0.031 | 0.086 | 0.026 | 0.012 | 0.035 | 0.004 | 0.162 | 0.237 | 0.825 | -0.241 | 0.305 | 0.041 | -0.208 | 0.323 | -0.035 | 0.475 | -0.197 | 0.001 |
| _cannot | 0.012 | 0.002 | -0.027 | -0.002 | 0.244 | 0.046 | 0.014 | 0.014 | 0.023 | -0.023 | 0.031 | 0.219 | 0.744 | -0.224 | 0.219 | -0.11 | 0.449 | 0.248 | 0.447 | -0.038 | 0.071 |
| _be | -0.008 | -0.005 | -0.082 | 0.011 | -0.128 | 0.006 | -0.038 | -0.003 | -0.003 | 0.025 | 0.012 | 0.09 | 0.242 | 0.454 | 0.776 | 0.894 | 0.105 | 0.062 | -0.037 | 0.135 | 0.039 |
| _implemented | 0.023 | 0.013 | 0.095 | 0.065 | 0.134 | -0.005 | -0.064 | 0.002 | -0.009 | -0.032 | -0.021 | 0.029 | 0.078 | 0.199 | 0.414 | 0.3 | 0.613 | 0.937 | 0.457 | -0.105 | 0.021 |
| . | 0.15 | 0.016 | 0.091 | -0.015 | -0.199 | -0.039 | 0.025 | -0.047 | 0.009 | -0.114 | 0.002 | 0.144 | 0.037 | -0.023 | -0.017 | 0.108 | 0.305 | 0.147 | -0.099 | 0.933 | 0.974 |
| </s> | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |