UVA DATA SCIENCE

DS 6051 Decoding Large Language Models

Introduction to Transformers

Chirag Agarwal
Assistant Professor
School of Data Science
University of Virginia

Recap

- Introduction to Tokenizers
- Byte-Pair Encoding
- Problem with Tokenizers
- Extending Tokenizers

The Early Ages - Prior to transformers

- Seq2Seq models - Sequence to Sequence
- Recurrent Neural Networks
- Long-Short Term Memory - LSTM
- Gated Recurrent Unit

Problems:

✘ Context : cannot understand

✘ Long term dependencies

https://education.nationalgeographic.org/resource/automobile/

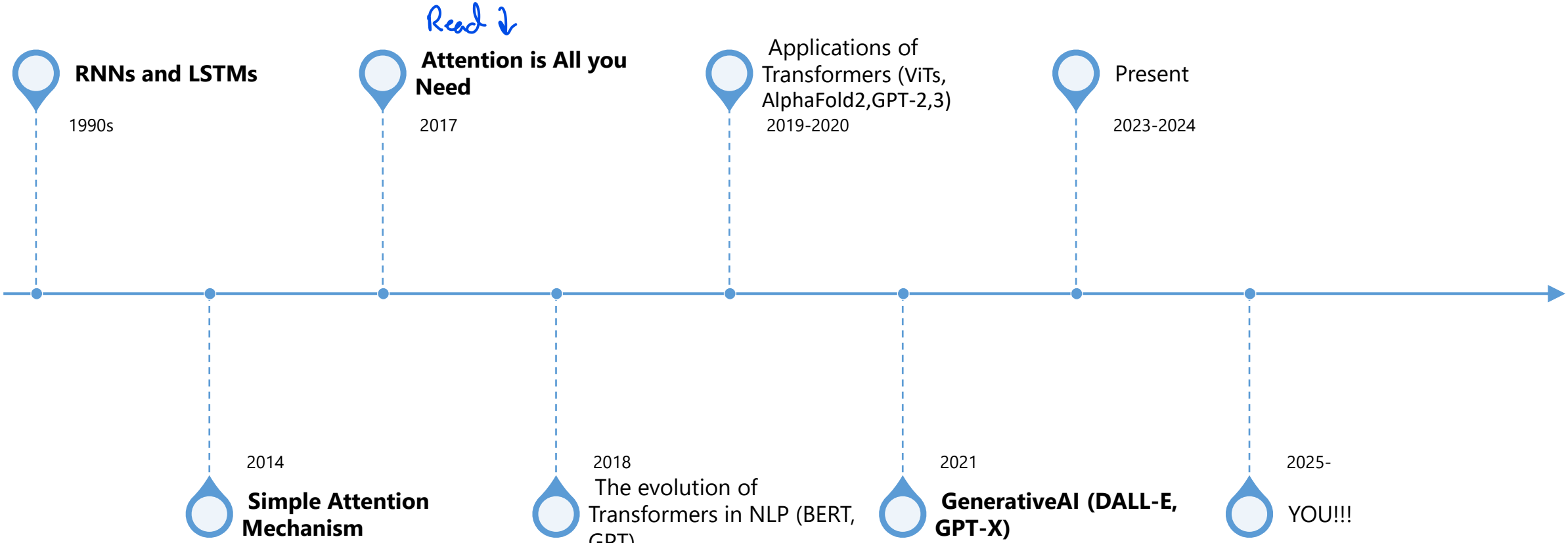
The Current Age

- Transformers – Solve the probs above
- Logical, Mathematical and Commonsense Reasoning
- Alignment using RLHF
  - ↳ align/match LLM to human principles (fairness/robustness)
- Safety testing and jailbreaks
- Ethical and Fair models

- ✓ Context
- ✓ Long term dependencies

https://www.teslarati.com/tesla-cybertruck-tour-japan/

Evolution of Attention



The Binding Model!!

Any data you can convert to a sequence you can use transformers on it!!!

↳ tokenize

A brief *mostly complete* history of Artificial Intelligence

starting around the cybernetics movement

From cybernetics to AI

The idea that humans and machines are essentially the same gives rise to a project where scientists began to seriously consider what it would take to develop machines with human-like intelligence.

The "birth of AI" in the wake of the cybernetics movement Bringing together the functioning of machines and organic beings

1940's

1942: Accelerated code breaking Alan Turing used the Bombe machine to decode messages encrypted using the Enigma machine at an accelerated pace during WWII

1949: "The Manchester Baby" runs its first program

1948: "Cybernetics" the study of control and communication in the animal and the machine by Norbert Wiener

1948: Mark 1 Prototype The first stored program computer also known as the "Manchester Baby" A proof of concept to test the first electronic random-access memory (RAM) device Built by Tom Kilburn & Freddie Williams

1949: "Giant Brains: Or Machines That Think" Edmund Berkeley compares machines to human brains if it were made of "hardware and wire instead of flesh and nerves."

1943: Machines and behavior "Behavior, Purpose, and Teleology" by Rosenblueth, Wiener, & Bigelow

1943: "Artificial Neurons" A Logical Calculus of the Ideas Immanent in Nervous Activity by McCulloch & Pitts

The birth of neural networks

1950's

1956: "AI will beat a human at chess within the next 10 years" -Herbert Simon

1955: First AI Workshop Proposed to be held at Dartmouth in 1956 McCarthy, Minsky, Rochester, Shannon

1950: THESEUS Robotic, maze-solving mouse that could "remember" its path. Claude Shannon, Bell Labs

1950: The Imitation Game Computing Machinery and Intelligence by Alan Turing

1957: The Perceptron learns to identify shapes; an early form of machine learning Created by Frank Rosenblatt

1958: LISP (List Processing) First Programming Language Created by: John McCarthy

1952: Program that can play Checkers is developed by Arthur Samuel*

1959: "Machine Learning" Introduced into the nomenclature by Arthur Samuel at IBM

1956: The first AI program Logic Theorist - prove theorems in symbolic logic from Whitehead and Russell's Principia Mathematica Created by Newell & Simon

1957: General Problem Solver Means-End Problem Solving Newell, Simon, & Shaw

1955: "Artificial Intelligence" Introduced into the nomenclature by John McCarthy

"The construction of computer programs that engage in tasks that are currently more satisfactorily performed by human beings because they require high-level mental processes such as: perceptual learning, memory organization and critical reasoning."
- Marvin Minsky

1960's

1970: "From three to eight years we will have a machine with the general intelligence of an average human being" -Marvin Minsky

1966: SHAKEY First robot to perceive its surroundings, navigate, plan a route, adjust for errors, and improve its abilities using English language communication Stanford Research Institute

1965: DENDRAL Expert System: hypothesis formation and science Inferences & Logical Conclusions Feigenbaum & Lederberg

1963: RAND First tablet and stylus created RAND Corporation

1964: STUDENT Lisp-based program that solved algebra word problems Created by Daniel Bobrow

1964: ELIZA Chatbot Natural Language Processing Created by Weizenbaum

1961: UNIMATE First Industrial Robotic arm available commercially Created by George Devol

1964: Facial Recognition Used "landmarking" techniques to identify facial features Created by Bledsoe, Wolf, & Bisson

1961: SAINT (Symbolic Automatic INTegrator) heuristic problem-solving program for symbolic integration in calculus Created by James Slagle

1968: SHDRLU Natural language processing program that controlled a block world using English language instructions created by Terry Winograd

1961: "The Stanford Cart" first autonomous vehicle created by James Adams

1968: The concept now known as "deep learning" is first introduced by Alexey Ivakhnenko in "Group method of data handling"

1968: Artist Vera Molnar uses punch card instructions to create digital art using the Fortran program- the first "dehumanized art"

1970's

1980-1987 **AI Boom**

1972: WABOT-1 The first "android" Walked, communicated in Japanese and gripped objects Waseda University

1973: Reduced funding for AI due to lack of promised progress in Britain

1979: Gammanoid beats champion at backgammon Created by: Hans Berliner

1976: Physical Symbol System Hypothesis "Computer science as empirical inquiry Symbols and Search" by Newell & Simon

1972: MYCIN "The granddaddy of expert systems" - Newell Simon Expert System: Blood Disease Diagnosis with a 65% accuracy Rate

1979: The Stanford Cart successfully navigates a room without human assistance

1975: SAM Program meant to "understand" stories run on scripts Created by Roger Schank

1971: AARON Autonomous Drawing Program creates the first "AI Art" - paints a "turtle" Designed by Harold Cohen

1974-1980 **1st AI Winter**

Perceptions failed to live up to expectations Discoveries made failed to make the promised impact

1971: Microprocessor Invented by Ted Hoff Ushering in the "Golden Age" of Expert Systems

Hoff received the national medal of technology & innovation in 2010

1980's

1987: VPL Data Glove A VR glove that controlled a virtual hand

1989: CYBERFACE VR glasses created LeepVR

1980: 1st National Conference on Artificial Intelligence Held at Stanford University, August 18-21

1981: 5th Generation Computer Project Japan spends $850 million to create computers that could translate and use human language + express human-level reasoning

1990: machine learning shifts from knowledge driven approaches to data-driven approaches

1988: Eigenface Linear Algebra is used to advance facial recognition programs Sirovich & Kirby

1985: RBX5 Using self-learning software the robot would advance from simple responses to making predictions about the future from past events RB Robot Corporation

1988: Shift from rule-based to probabilistic methods of machine translation IBM T.J. Watson Research Center

1983: The problem of "knowledge acquisition" acknowledged as a barrier to AI progress

1984: "Winter is Coming" -Schank & Minsky

1985: Teddy Ruxpin The World's First Animated Talking Toy. Created by Ken Forsse

1987: A new architecture of cognition "Parallel Distributed Processing" Feldman, Hayes, McClelland, Rumelhart

1988: JABBERWACKY Amusing chatbot Created by: Rollo Carpenter,

1980: XCON First expert system on the commercial market Used to configure customer orders and reduce shipping times

1980: WABOT-2 Humanoid robot that communicated with people, read music, and played music on an electronic organ Waseda University

1986: Back-Propogation for networks of neuron-like units "Learning representations by back-propogating errors" by Rumelhart, Hinton, & Williams

1986: First Driverless Car Could drive up to 55 mph

1987-1994 **2nd AI Winter**

Limitations of if-then reasoning apparent Expert systems based on if-then reasoning failed because software was hard to maintain and couldn't handle novel information

1989: First successful application of the backpropogation algorithm Used to recognize handwritten zip codes AT&T Bell Labs

1989: The terms "Hyperlink" and "hypertext" are coined by Tim Berners-Lee

1987: Market for specialized LISP-based hardware collapses Low consumer, public, and private interest in AI

1990's

2006: "Deep Learning" coined by Geoffrey Hinton to explain new algorithms that can be trained to recognize objects and text in images and videos

1997: LSTM RNN used for handwriting and speech recognition Created by Hochreiter & Schmidhuber

SOCIAL MEDIA

1995: A.L.I.C.E. First chatbot that also collected data

1997: Six Degrees First social media platform

1998: FURBY First pet robot By Caleb Chung

1994: GPU 3D graphics introduced with the Play Station 1 home gaming console

1989: The "web" is invented by Tim Berners-Lee

world wide web

Internet Boom

1990: WorldWideWeb The first web browser is launched Created by Tim Berners-Lee

1997: Speech Recognition Software

1992: TD-Gammon Artificial Neural Net trained to play backgammon IBM

1997: Deep Blue beats Grand Master at Chess IBM Expert System that used a systematic brute force system

41 years after Simon's prediction

1998: COMPAS Software that assigns an algorithmic score for recidivism risk is used across several states to determine sentencing In 2016, it was shown that the algorithm was giving biased assessments based on race

1997: KISMET "Lifelike" social robot inspired by child developmental psychology that could recognize and display emotions

"BIG DATA" Introduced into the nomenclature

1993: FERET Program Developed to create algorithms for automatic face recognition software and to create a database of faces

1990 "AUGMENTED REALITY" Introduced into the nomenclature by Tom Caudell

2000's

2002: ROOMBA i-Robot

2002: Digital information storage surpasses non-digital information storage

2007: UX Algorithms Used to enhance advertising and user experience

2003: Mars Rover Navigates Mars without human intervention

2007: ImageNet Visual Database designed to use in visual recognition software research 20,000 categories

2004: BigDog Boston Dynamics creates dynamically stable quadruped military robot

2000: da Vinci First robot surgical system

2005: Mechanical Turk Service that recruited humans to hand label over 3 million images across 5k categories for computer vision tasks

2004: STANLEY first autonomous vehicle to win the Defense Advanced Research Projects Agency's 'Grand Challenge' by successfully navigating a 132-mile course in the Mojave Desert on it's own Stanford AI Lab

2010's

2014: Generative Adversarial Networks produce completely new images for the first time rather than working on pre-existing images. developed by Ian Goodfellow

2017: Google Deep Mind AI avatar teaches itself how to walk

2010: XBOX 360 Kinect First gaming hardware to track body movements and translate it into gaming directions

2011: Video Recognition Google X uses a neural network of 16,000 computer processors with 1 billion connections to browse YouTube. The AI identifies cats as a common image.

2013: Word2Vec program that converts language into math is created by Google In 2015 it was discovered that the model produced biased results such as refusing to categorize women as doctors generating concerns about biased training data

2016: Google AI's AlphaGO beats Lee Sedol at GO

2011: WATSON created & wins on Jeopardy!

2013: Atlas First iteration of the humanoid robot is created for search and rescue tasks Boston Dynamics

2017: AlphaGo Zero beats AlphaGo using a self-training model

2019: AlphaStar Beats Grandmaster at StarCraft2

2011: SIRI Apple releases virtual assistant

2015: DeepDream Convolutional Neural Net that using deep learning produced hallucinogenic imagery

2011: WildCat Boston Dynamics creates the fastest quadruped robot on Earth

2016: SOPHIA Humanoid Robot becomes the first "robot citizen" Created by David Hanson Jr.

2010: "AlexNet" Convolutional Neural Network achieves 15.3% error rate with image recognition using deep learning In 2015 it was discovered that the AI mislabels images featuring people with dark skin as "Gorilla" As a solution, Google+ removes the Gorilla label entirely. As of 2023, the label is still removed from the database

2018: Language processing AI beats human intellect on a Stanford reading and comprehension test for the first time Created by e-commerce giant Alibaba

2016: The Next Rembrandt Deep learning algorithm creates realistic portrait style painting using 3D printing

2010: MASSIVE DATA Access to massive volumes of data made training systems on image classification and recognition possible

2017: Dialog Agents Chatbots programmed in English and designed to negotiate with each other invented their own language Facebook AI Research Lab

2015: SWARM AI A real-time online tool that makes predictions based on converging information from various sources within a network predicted the winning horse at the Kentucky Derby Unanimous AI

2020's

2023 The Rise Of GENERATIVE AI

2020: GPT-3 Uses deep learning to create code, poetry, and other language writing tasks. OpenAI

2023: GEMINI First AI to outperform humans at massive multi-task language understanding tasks. Google DeepMind

2020: AlphaFold OpenAI deep learning system identifies three-dimensional structures of proteins

2023: GPT-4 The next iteration of OpenAI's GPT system A large multimodal model that accepts images and texts as inputs. Passed the Bar Exam scoring in the top 10%

2023: Alibaba's Iimproved AI Updated version that can interpret images, carry on complex conversations, and answer open ended questions

2018: BERT First bidirectional unsupervised learning model for natural language tasks Created by GOOGLE

2021: DALL-E OpenAI's diffusion model generative AI that can process images to create accurate captions

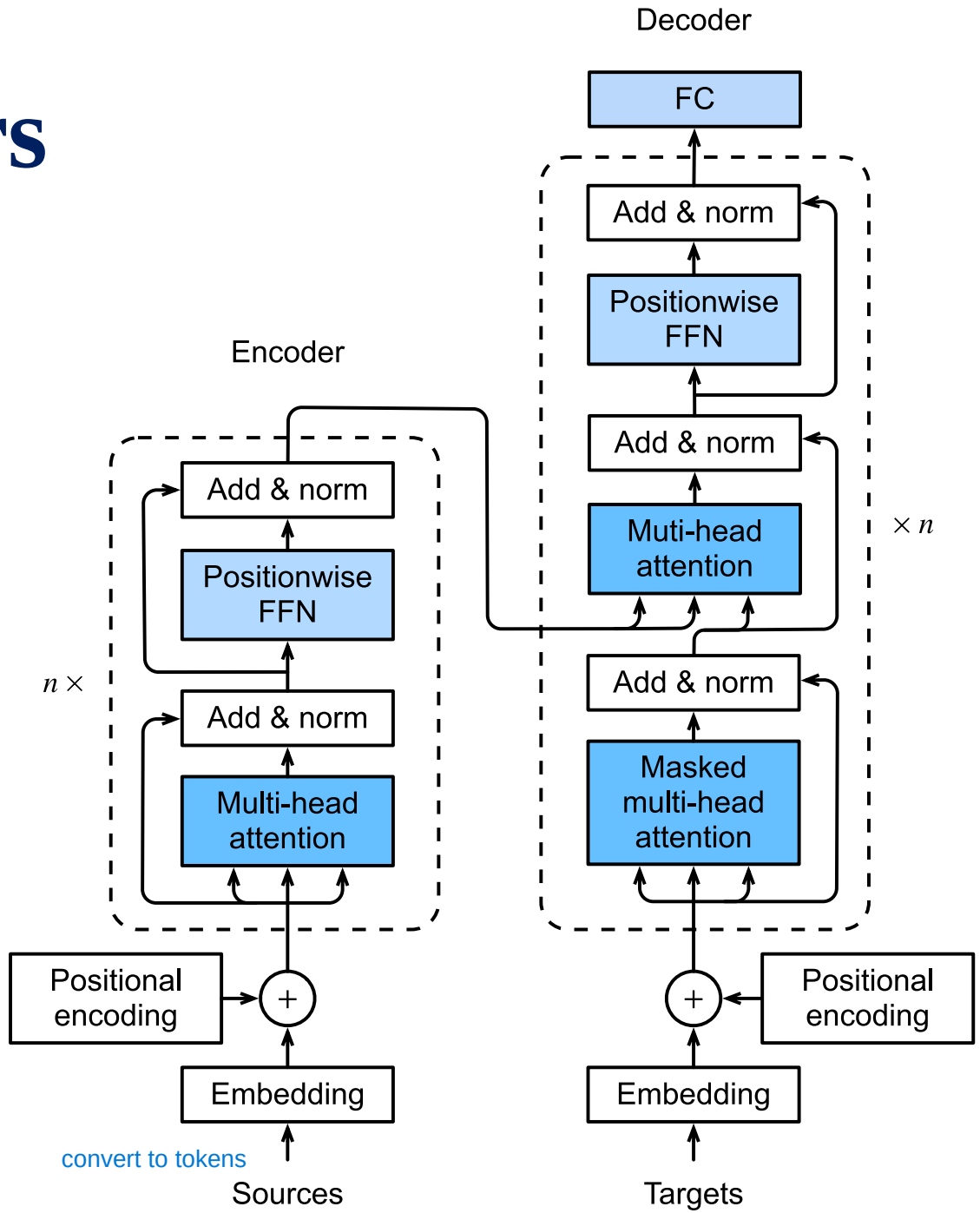
Information collected and poster designed by Danielle J. Williams, PhD | Washington University in St. Louis
daniellejwilliams.com

*some sources report the checkers program was developed in 1959

Transformers

Attention Is All You Need

Image →



Input Processing

1st task is to tokenize

- Input
- Embedding Layer
- Position Embeddings

Let's consider a Dialogue Completer

Input Dialogue

It is our choices, Harry, that show what we truly are,

If you want to know what a man's like, take a good look at

It matters not what someone is born,

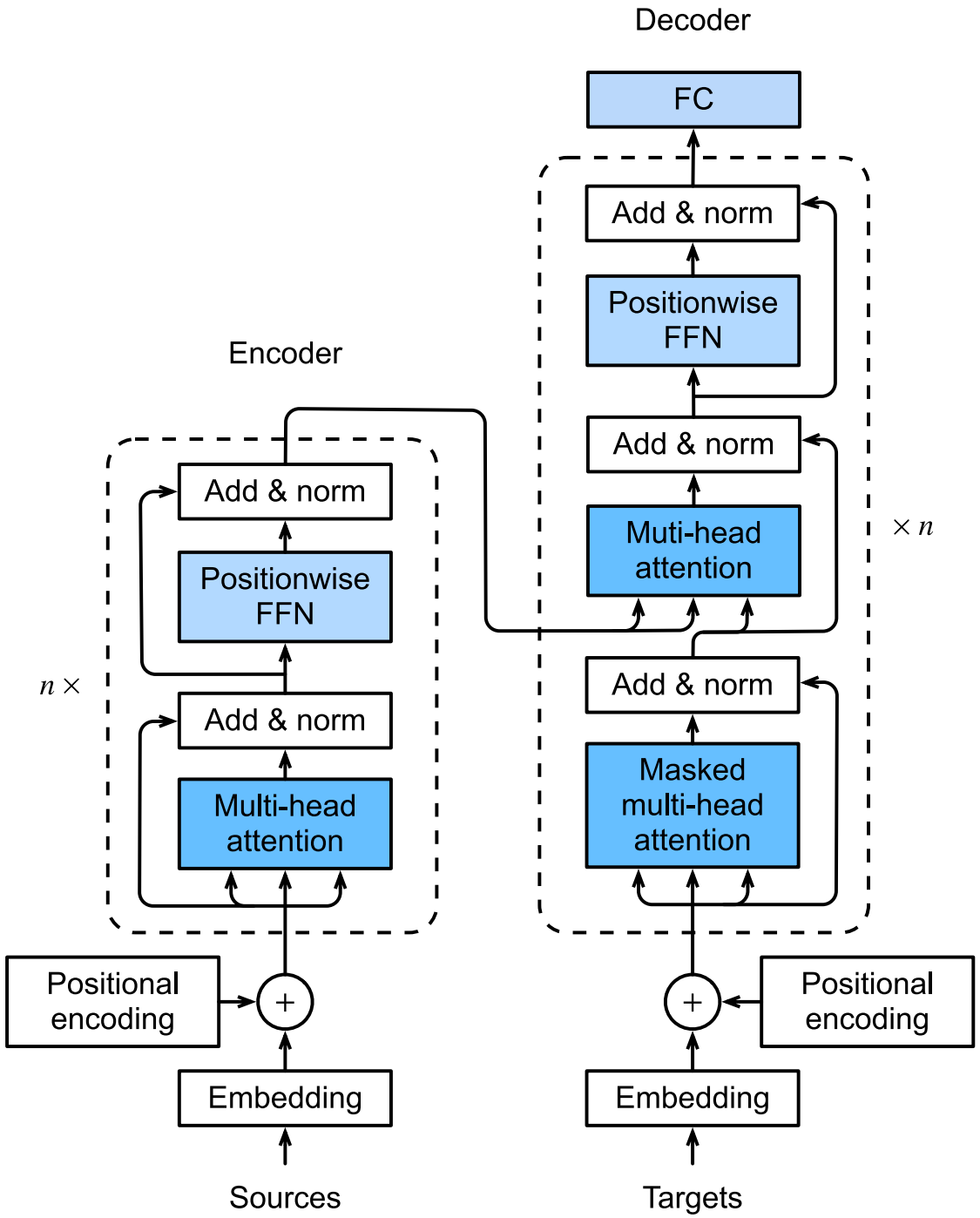
Dialogue Completion

<start> far more than our abilities <end>

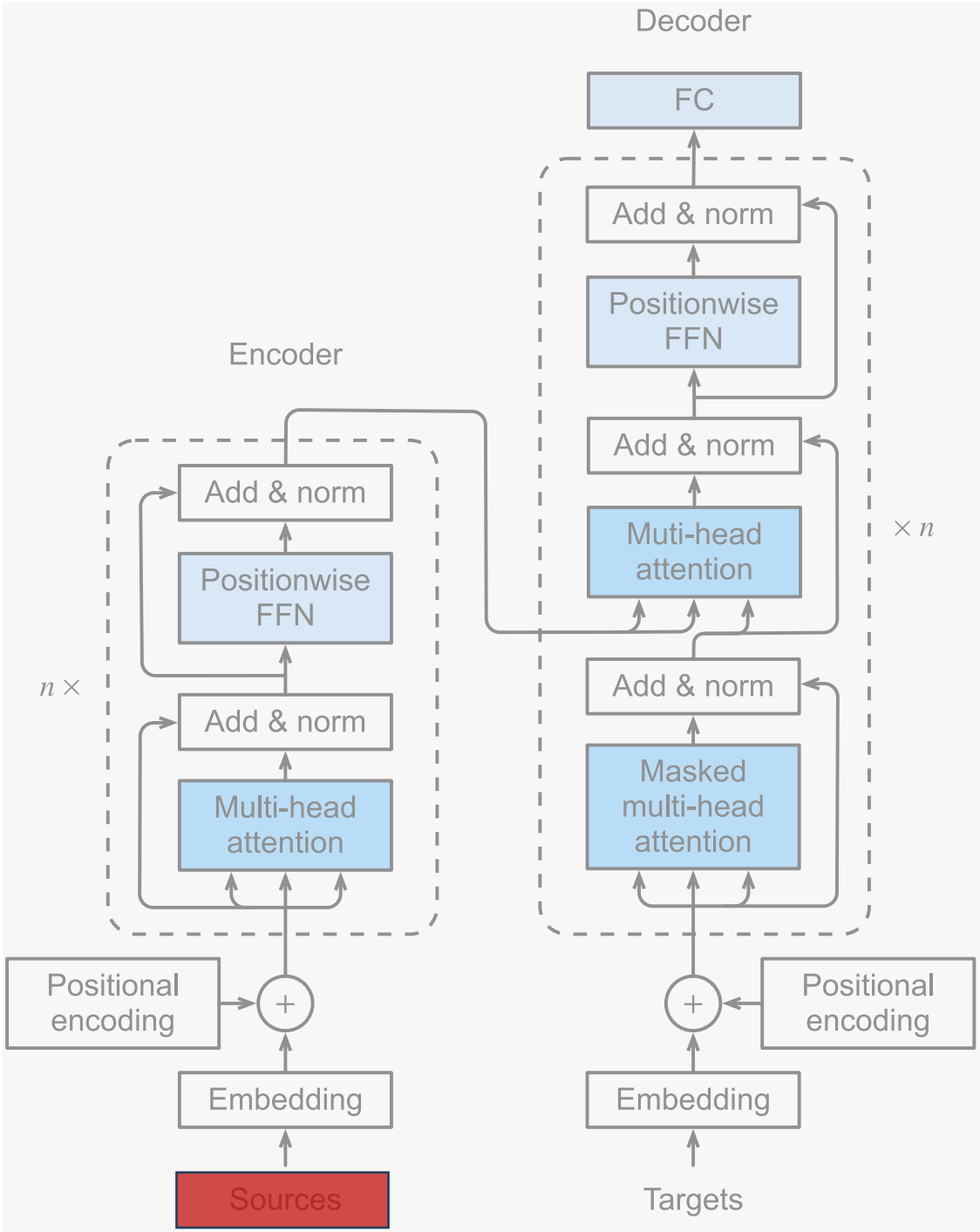
Prompts

<start> how he treats his inferiors, not his equals <end>

<start> but what they grow to be <end>



https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



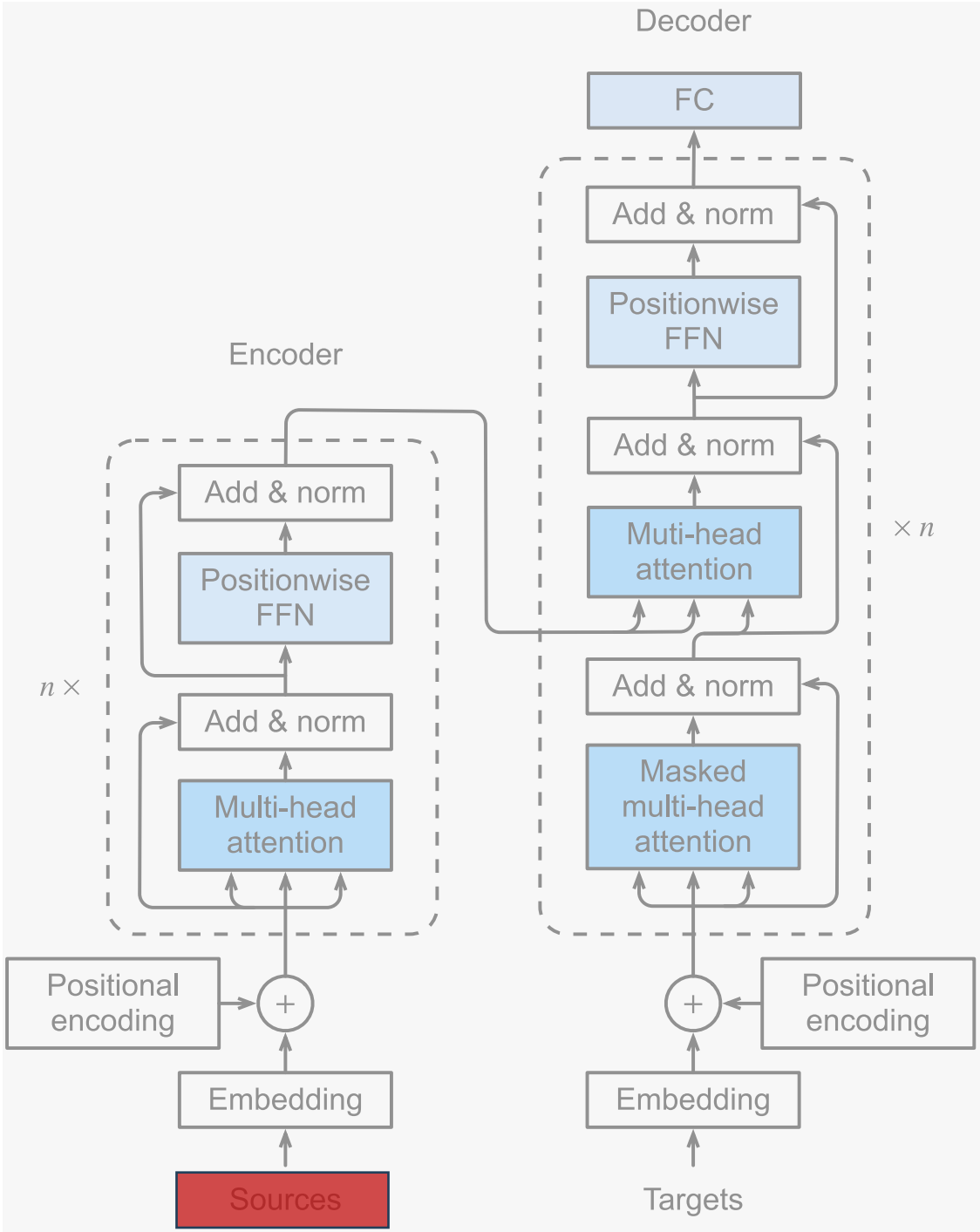
Input

Text

It matters not what someone is born,

ignore

https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



- First LLM is given a vocab to recognize

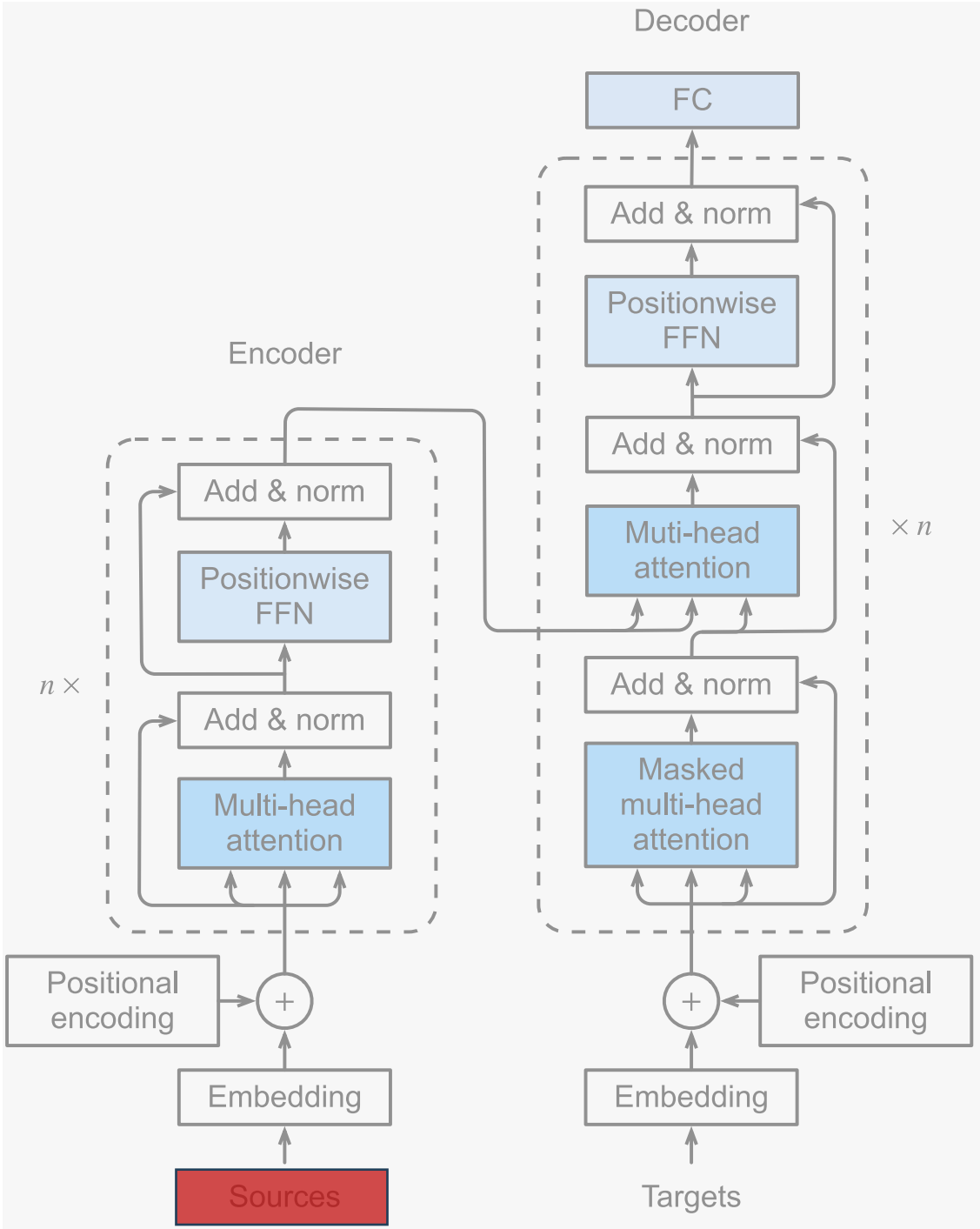
Input

Text: It matters not what someone is born,

Vocabulary	Indices
…	…
Harry	0
It	1
matters	2
Ron	501
not	612
what	697
…	…
someone	891
<YouKnow	1111
Who>	
is	1200
born	1893
Hermione	3019
,	5001

of those words in the vocab, retrieve the tokens of words we input

https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html

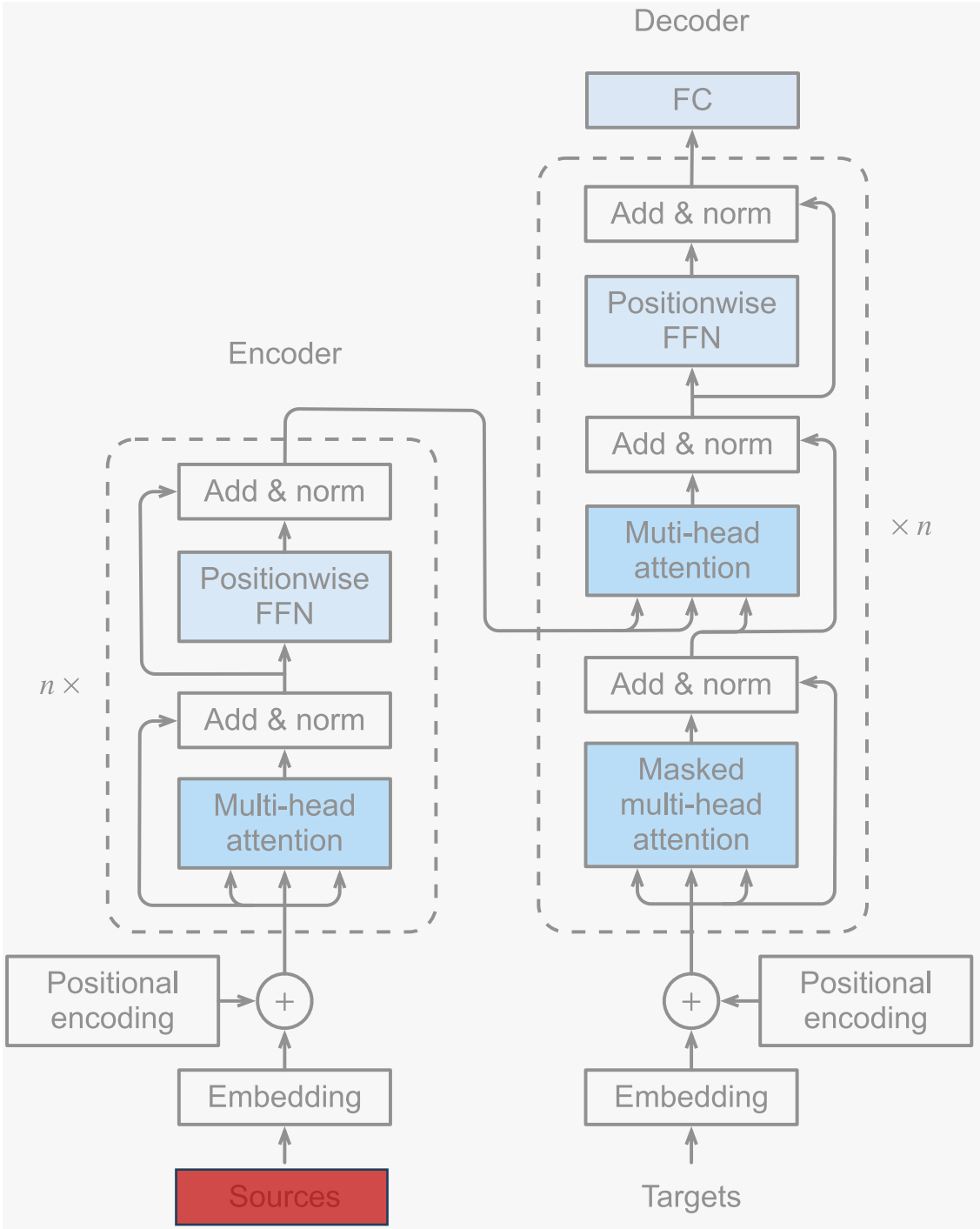


Input

Text: It matters not what someone is born,

Vocabulary	Indices
It	1
matters	2
not	612
what	697
someone	891
is	1200
born	1893
,	5001

https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



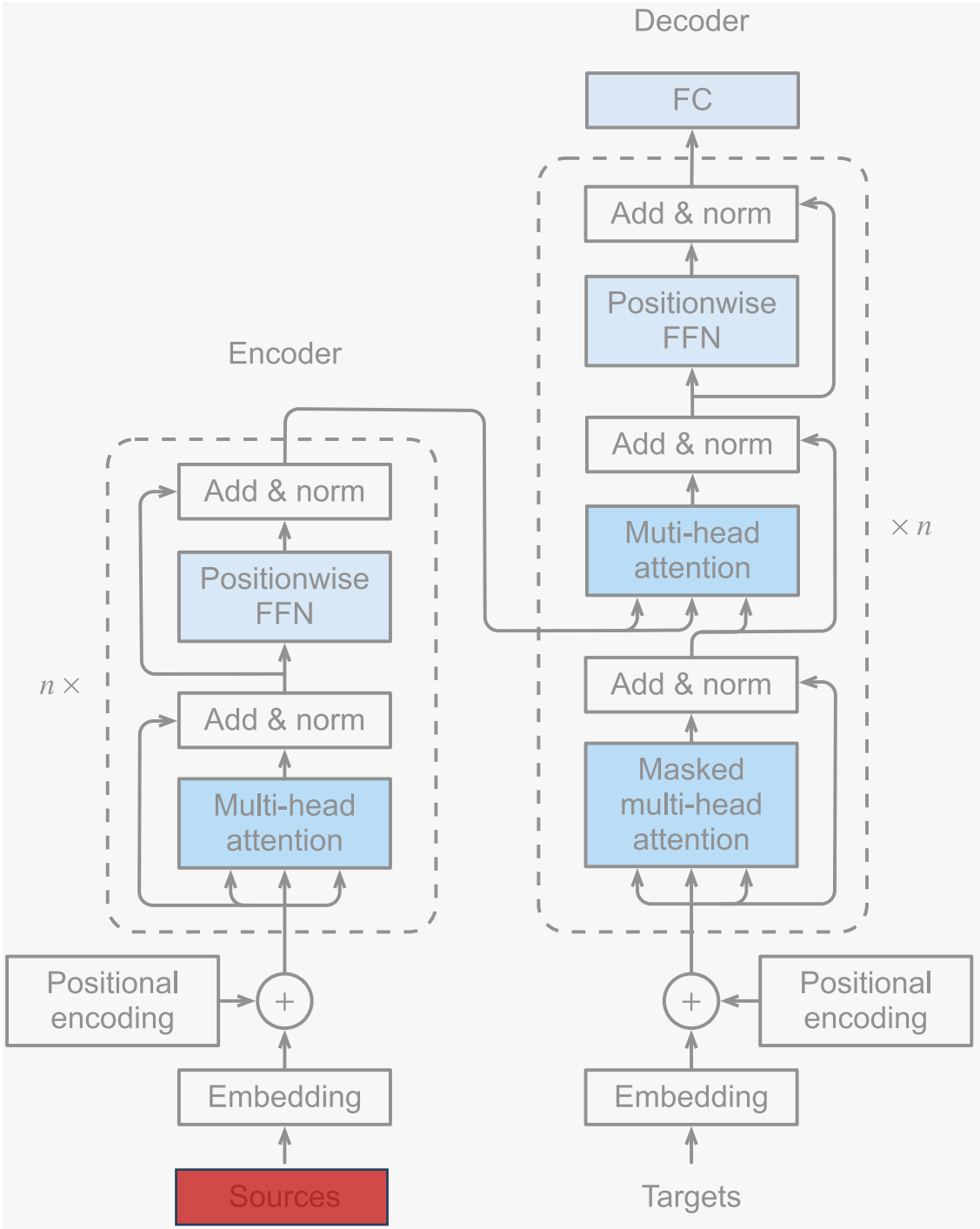
Input

Text It matters not what someone is born,

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Vocabulary Indices	1	2	612	697	891	1200	1893	5001

Now that is the input sample of indices, exact location of those words in the vocab it has
This is passed to the embedding layer

https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



Input

Text: It matters not what someone is born,

Vocabulary Indices: [1 2 612 697 891 1200 1893 5001]

V is vocab amount
D dimensions are what it is scaled down to

embedding layer 50K X 128
but we don't have to send the whole 50K we can send a subset of just relevant to the input

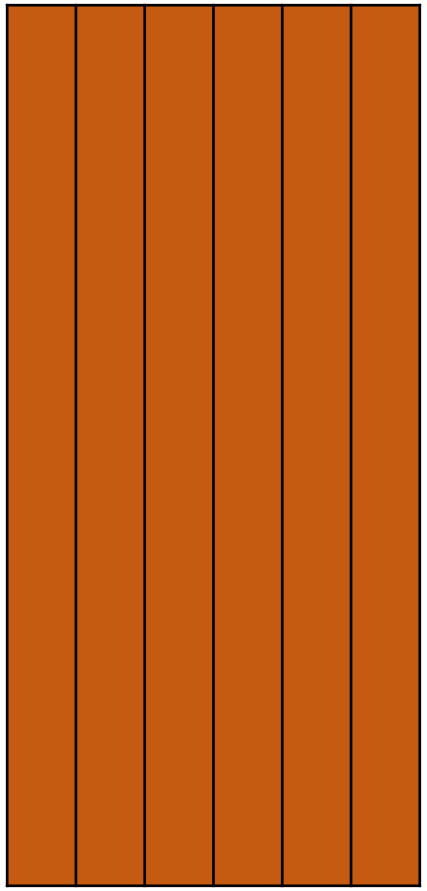
Input to be passed to the embedding layer

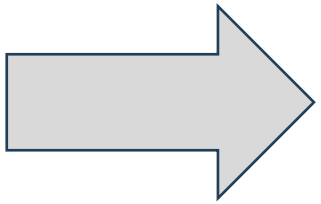
https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html

Hidden Layer Weight Matrix

100 hidden layer neurons

50,000 vocabulary words

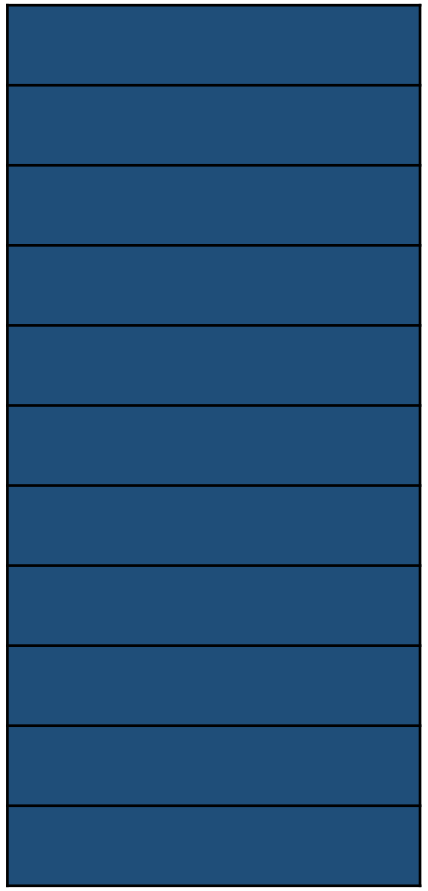


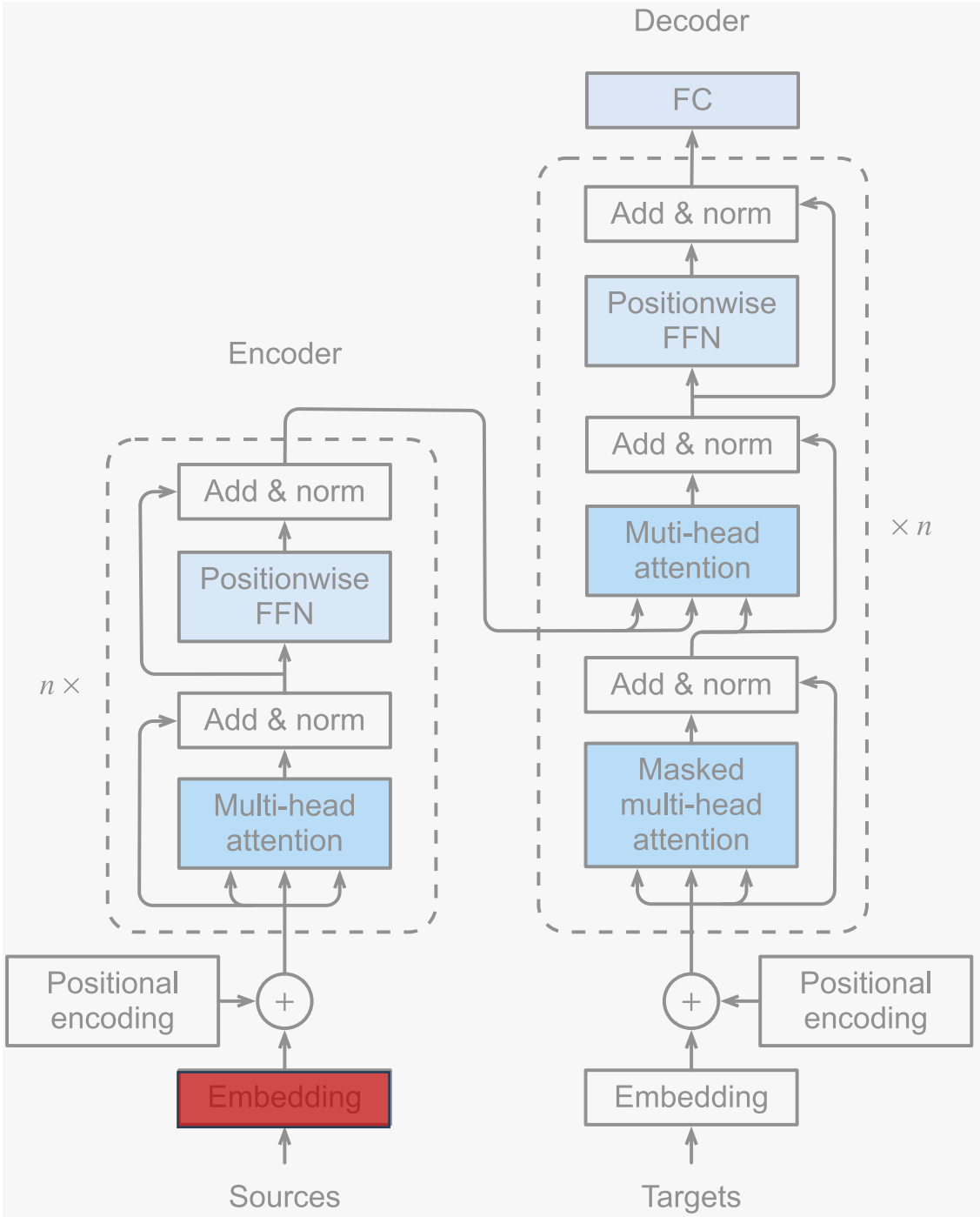


Word Embedding Dictionary

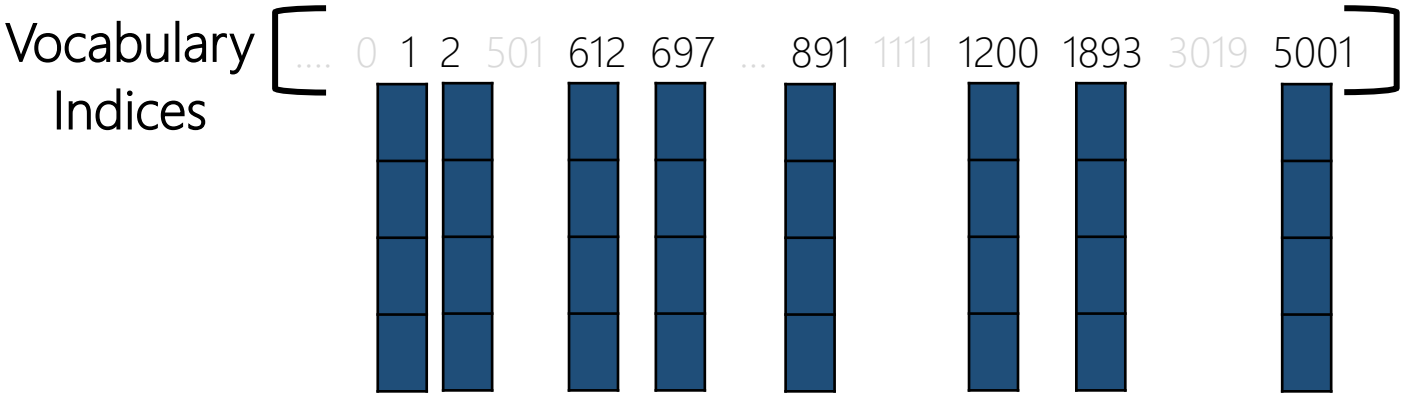
100 embedding features

50,000 vocabulary words



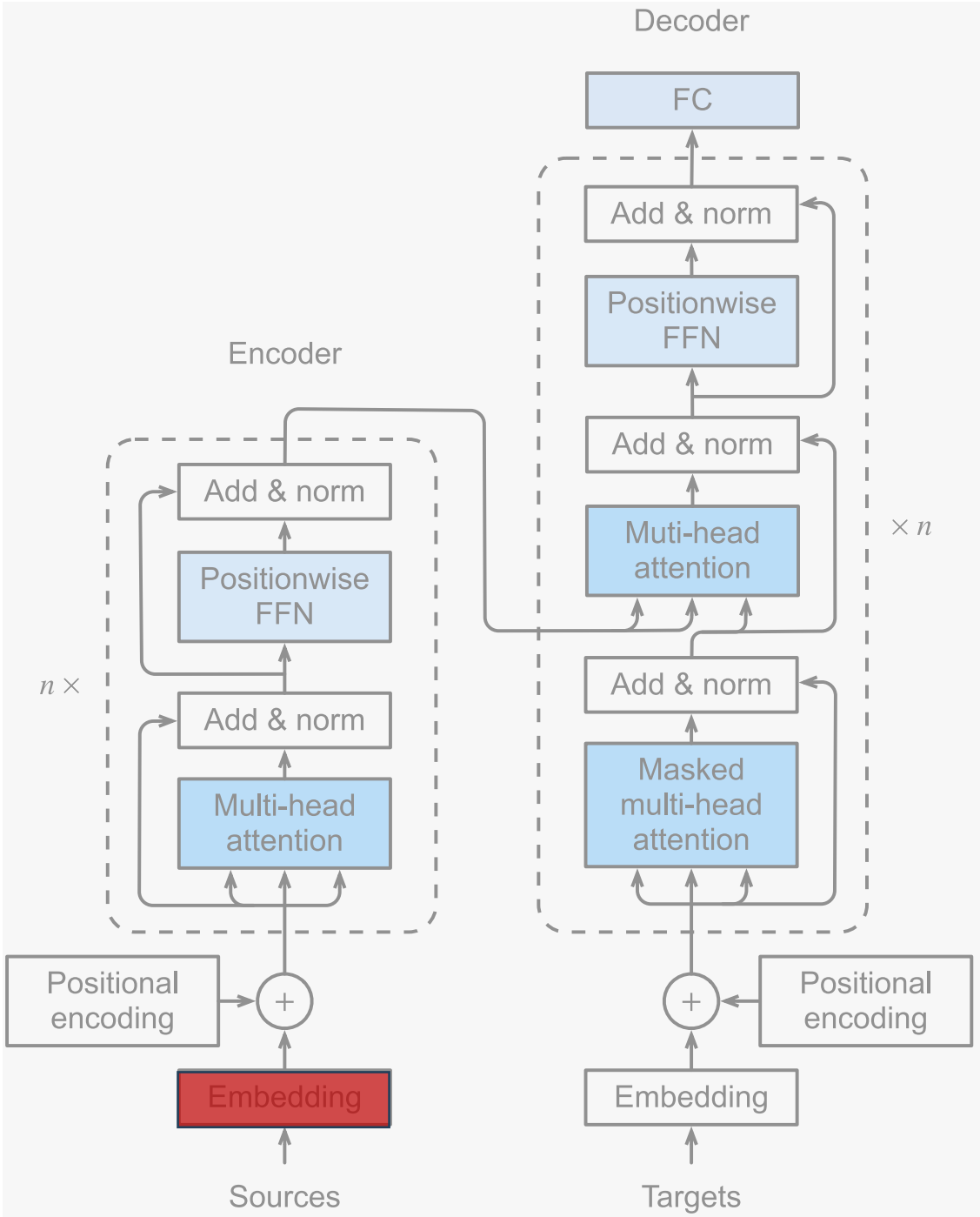


Input Embeddings



Input now 7x4

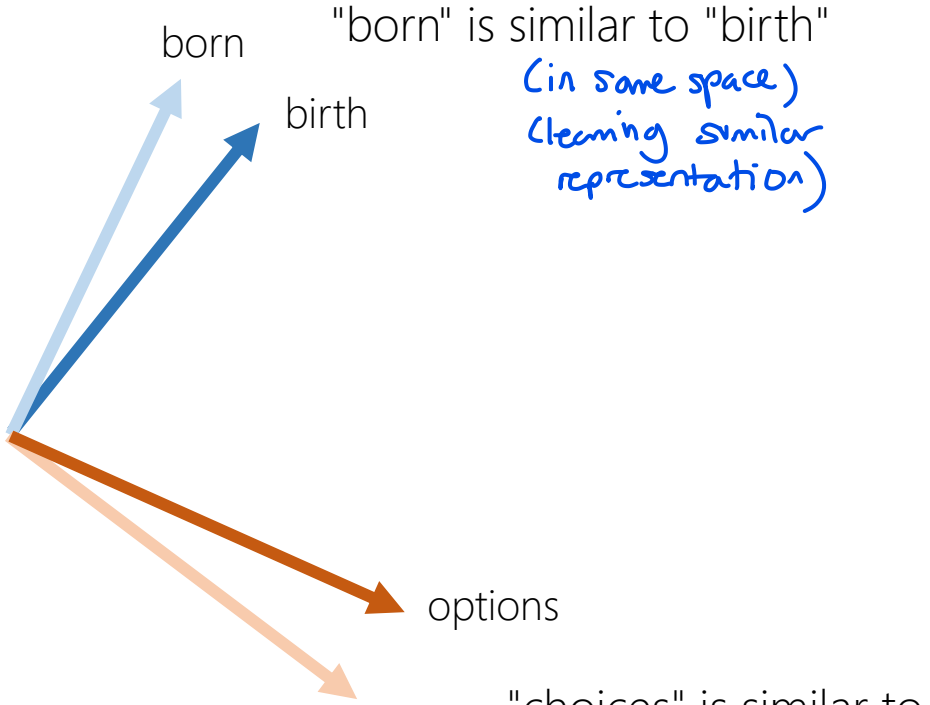
https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



Input Embeddings

Vocabulary Indices

....	0	1	2	501	612	697	...	891	1111	1200	1893	3019	5001
											born		

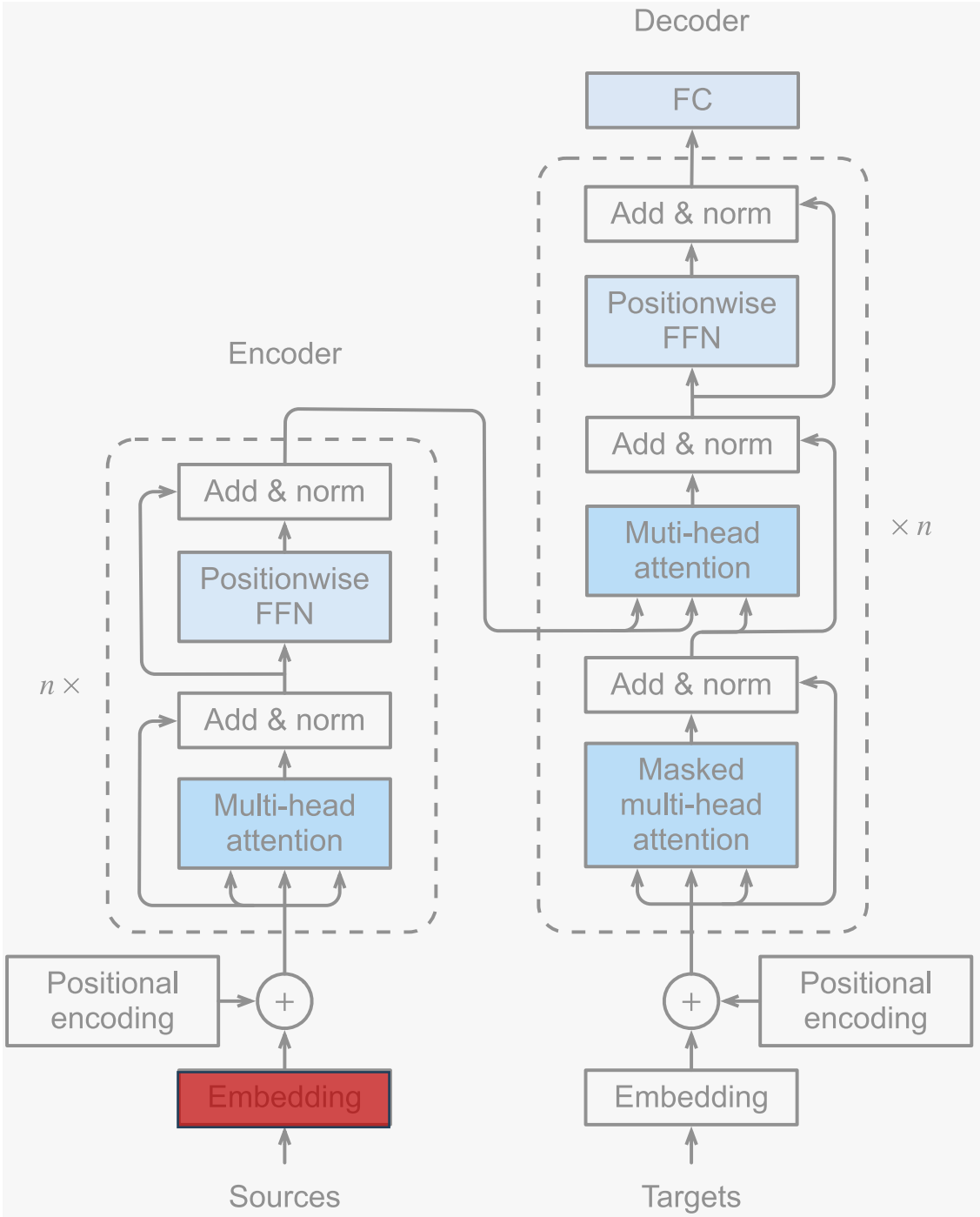


"born" is similar to "birth"

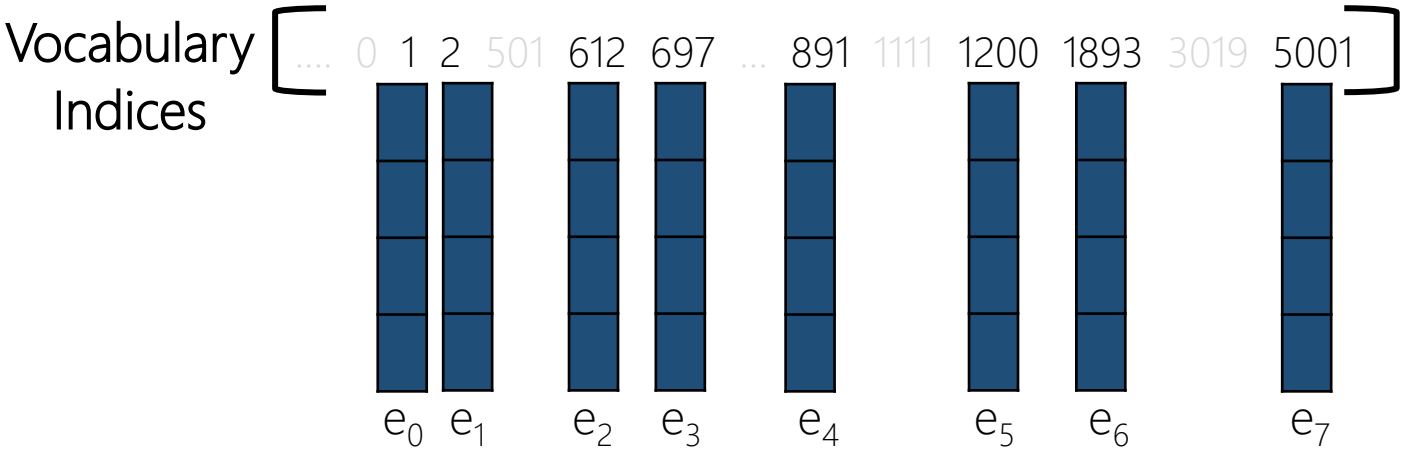
(in same space)
(learning similar representation)

"choices" is similar to "options"

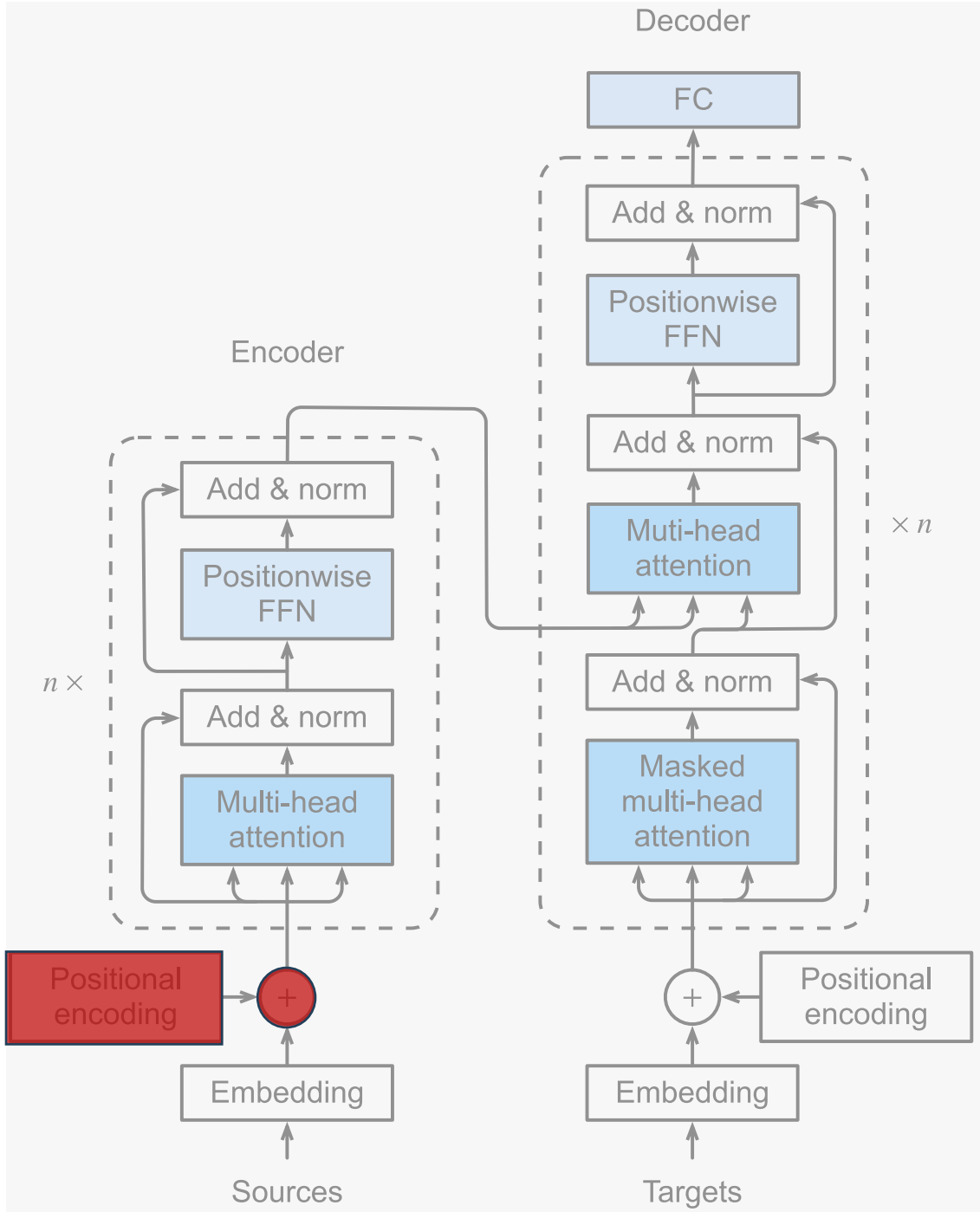
https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



Input Embeddings



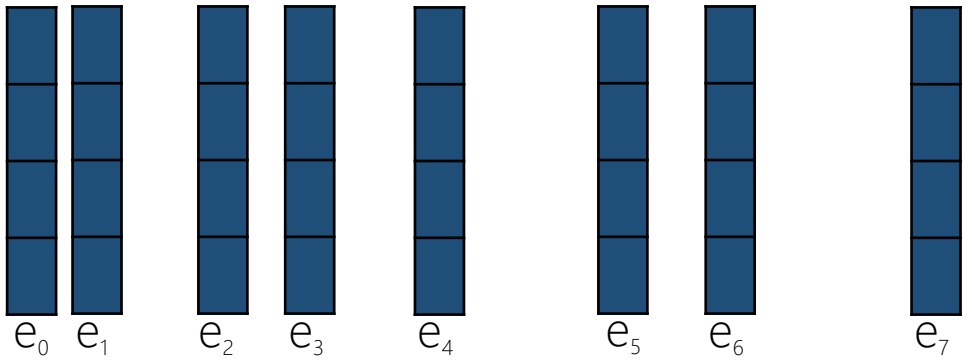
https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



Why do we need Positional Embedding?

we don't send sequentially, we process in parallel, might miss context/position of words close by

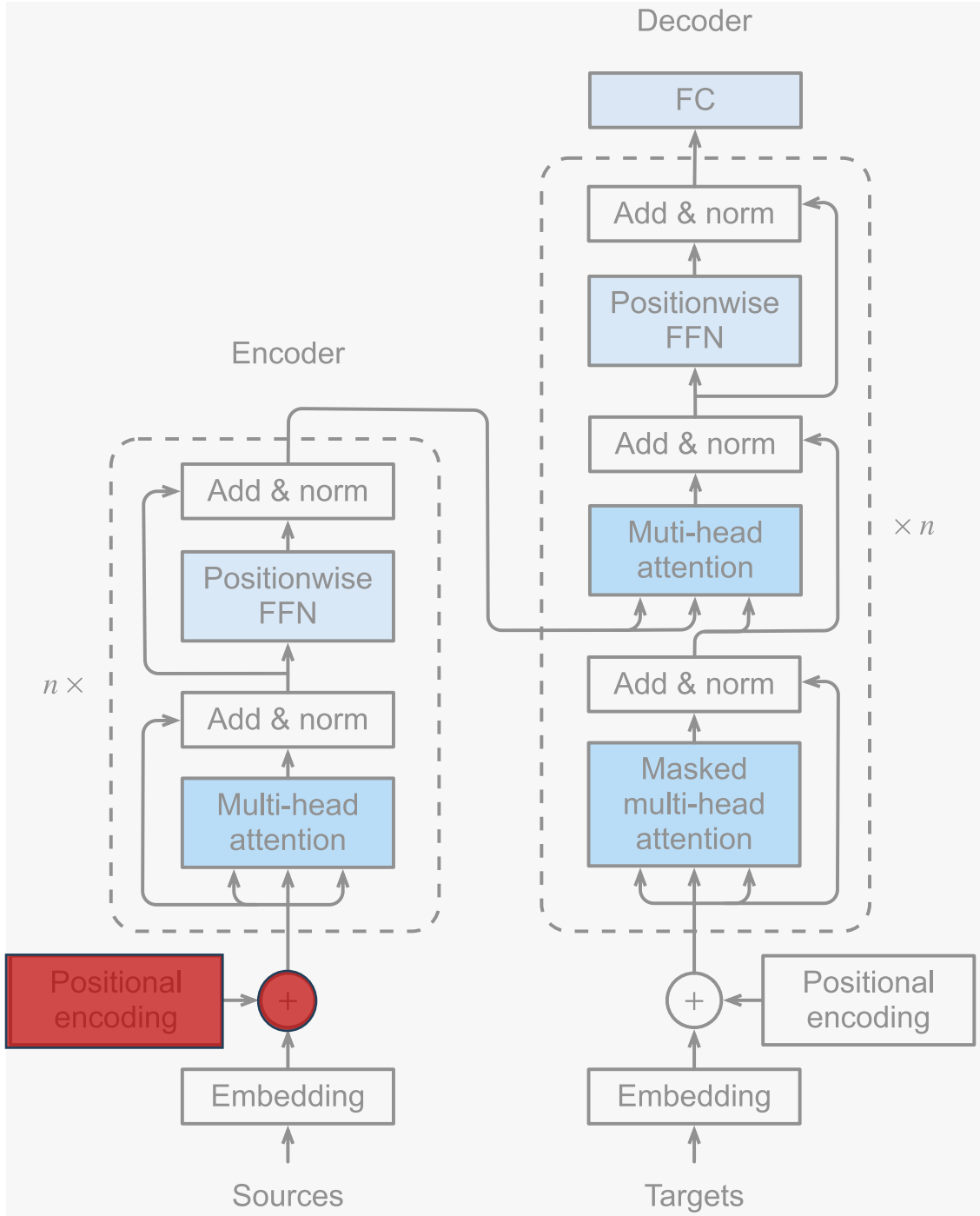
arbitrary 4D but ideally 128 or 256 dimensions



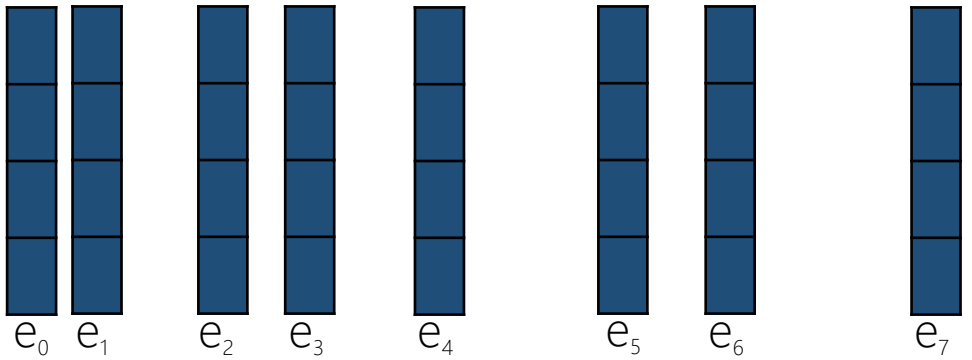
Positional encoding informs the transformer of the order of the words to not loose that info

Recurrent Neural Network
or
Long-Short Term Memory

https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html

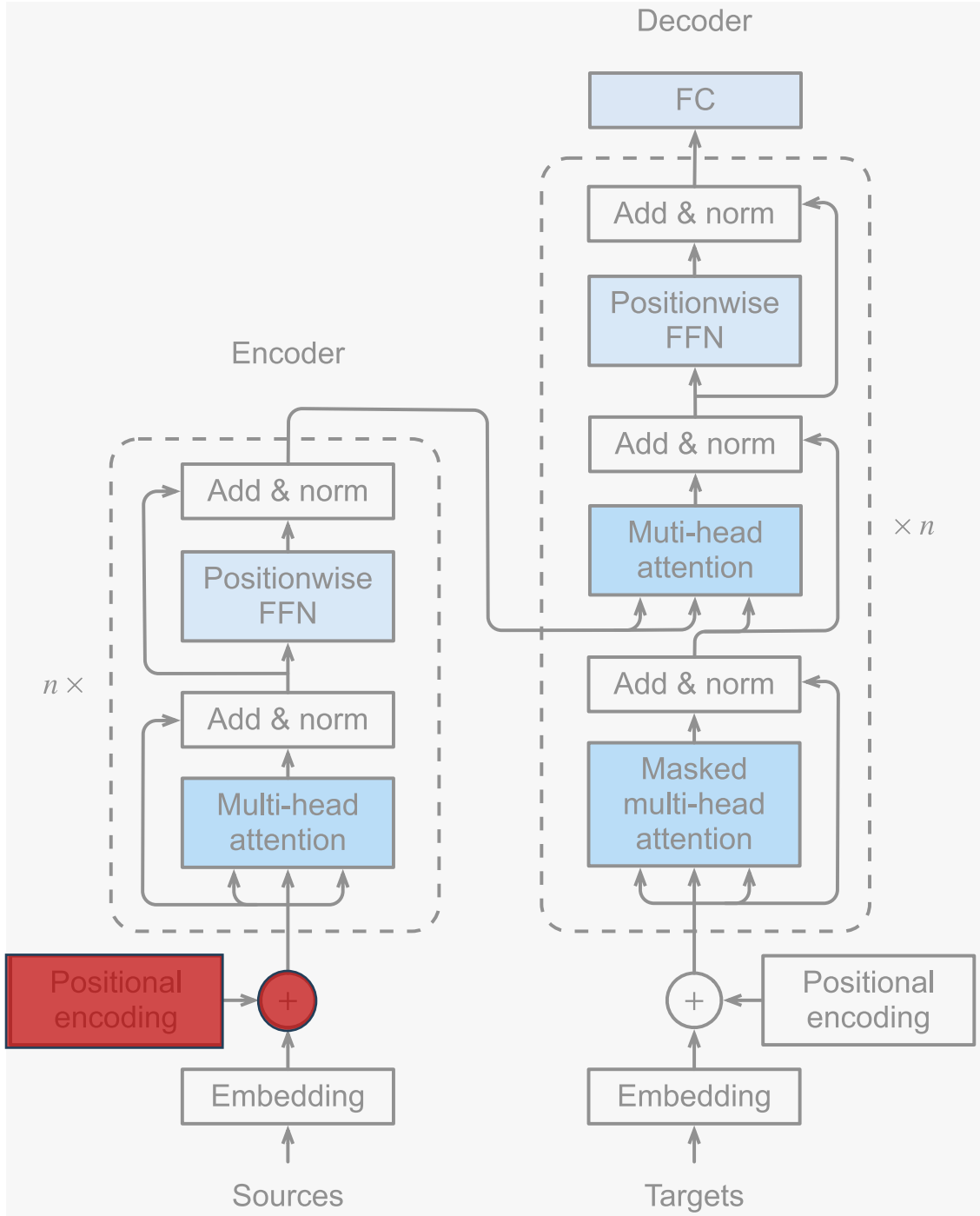


Why do we need Positional Embedding?



Transformers

https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



Why do we need Positional Embedding?

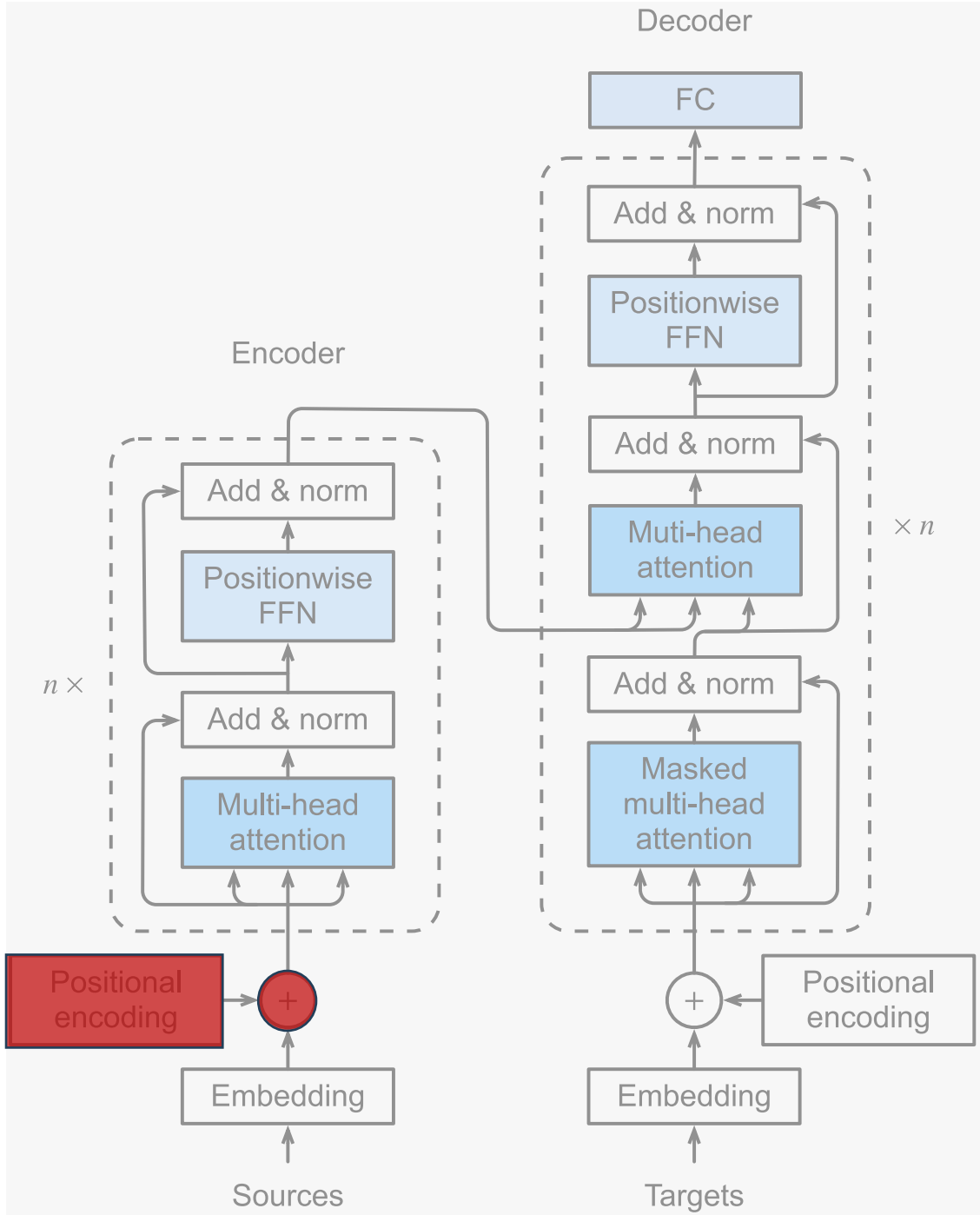
Importance of positon

Shows why we need the word order to noted

England invented Football, which is now played in 211 countries, including the USA, Canada, and Brazil.

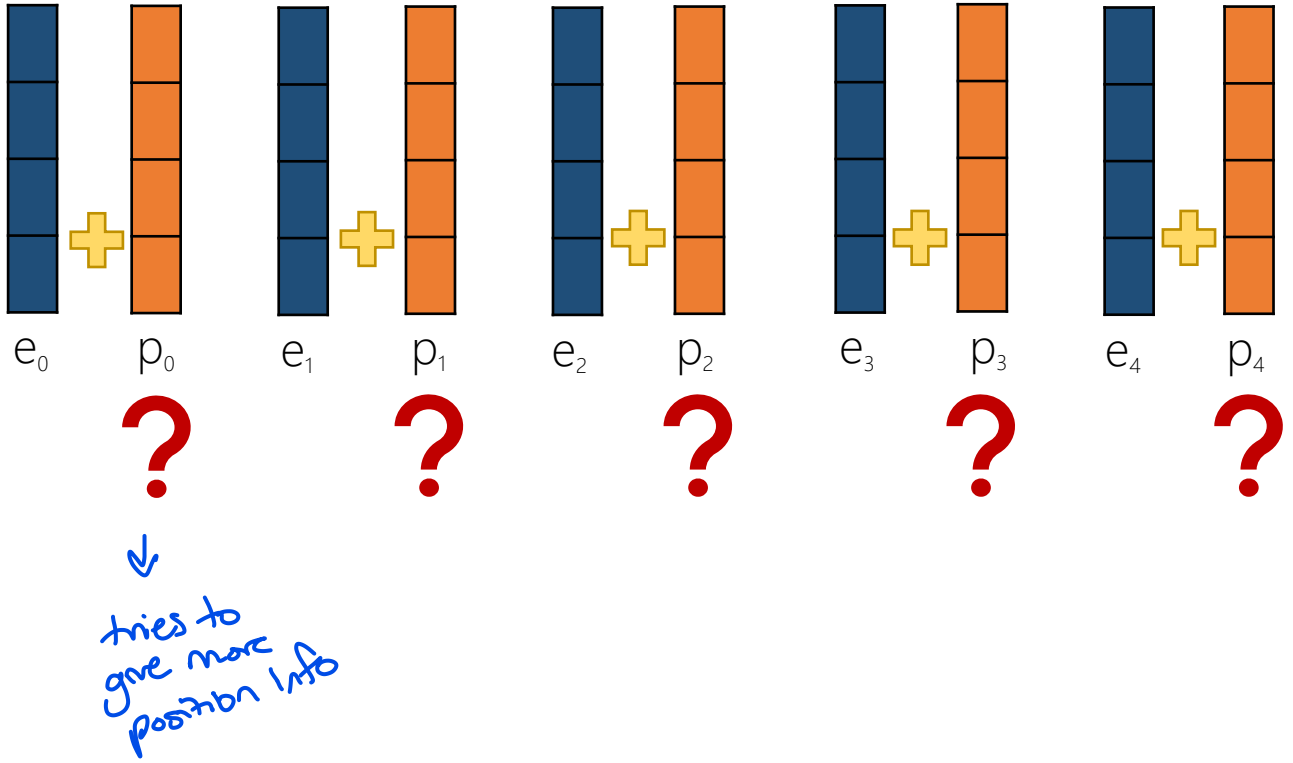
Brazil invented Football, which is now played in 211 countries, including the USA, Canada, and England.

https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html

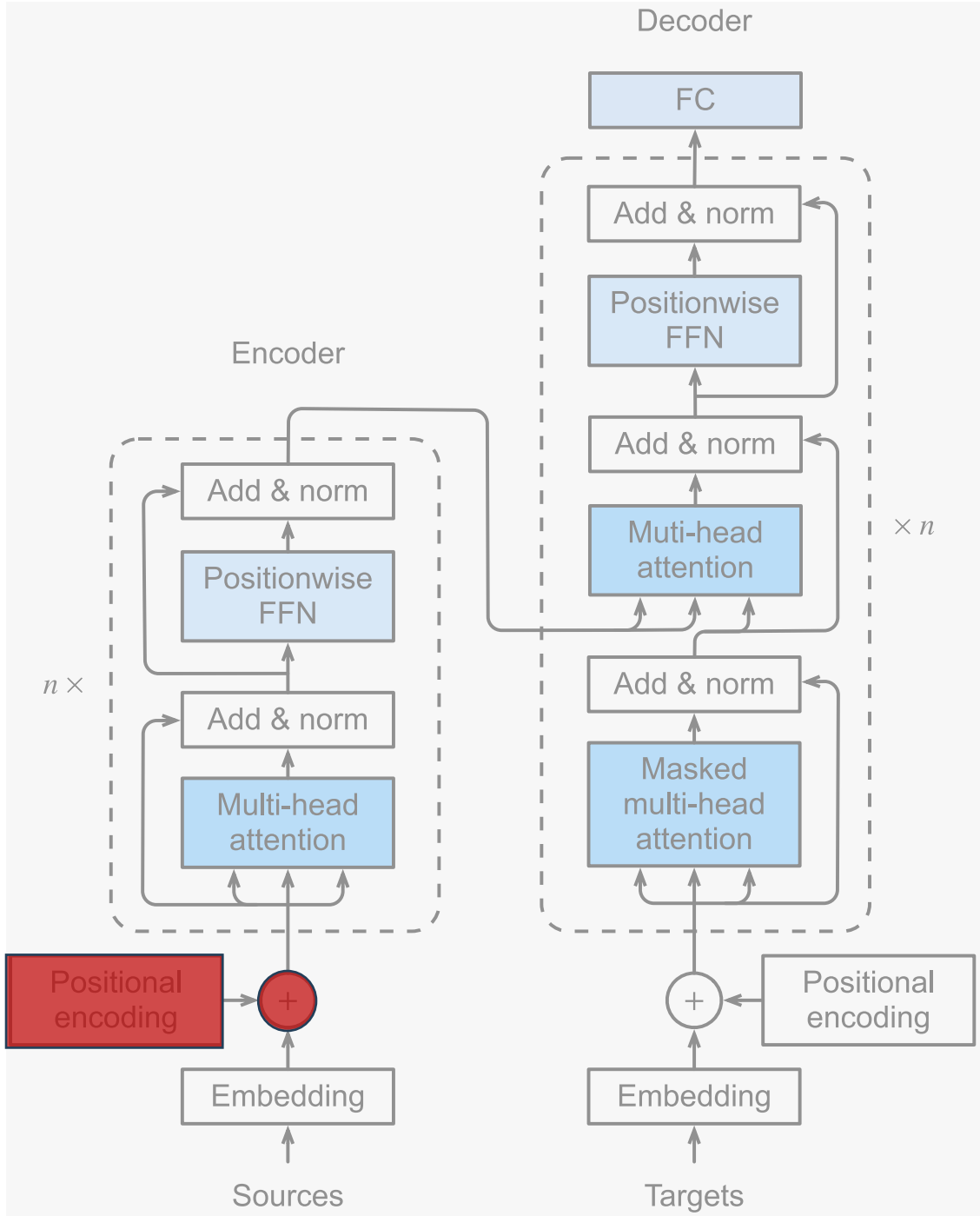


What to use as Positional Embedding?

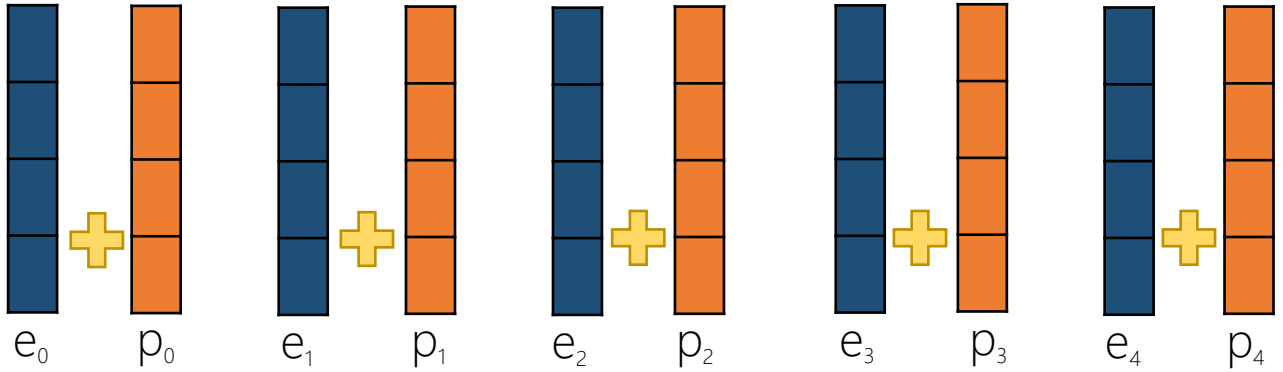
Cannot just use a vector of positions because like some prompts are 1000 words and this is too much for the LLM to process, so we normalize and note position more smartly



https://d2l.ai/chapter_attention-mechanisms-and-transformers/transformer.html



Frequencies for Positional Embedding



This nomalizes position

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

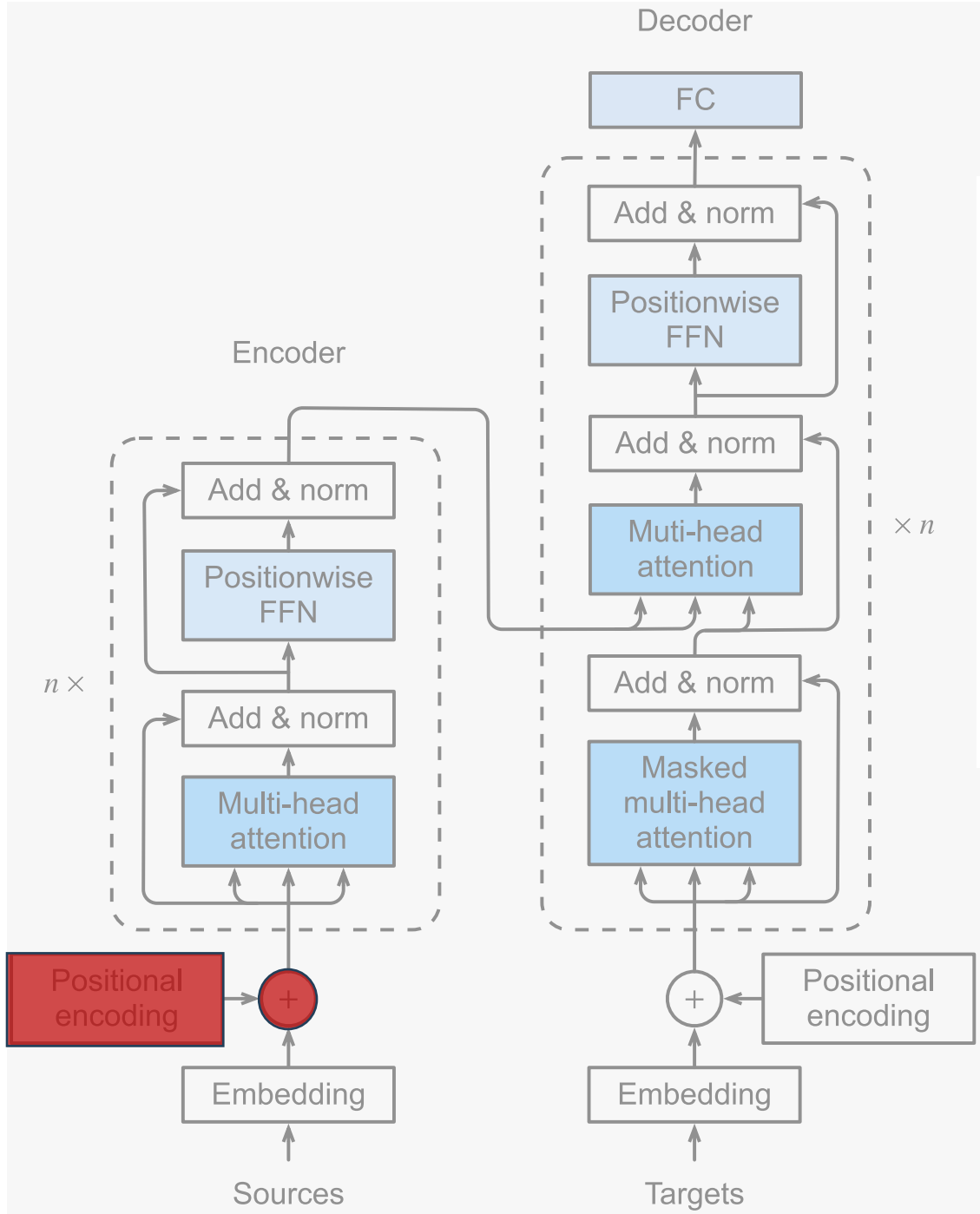
location of word in sentence

dimension- 4 here

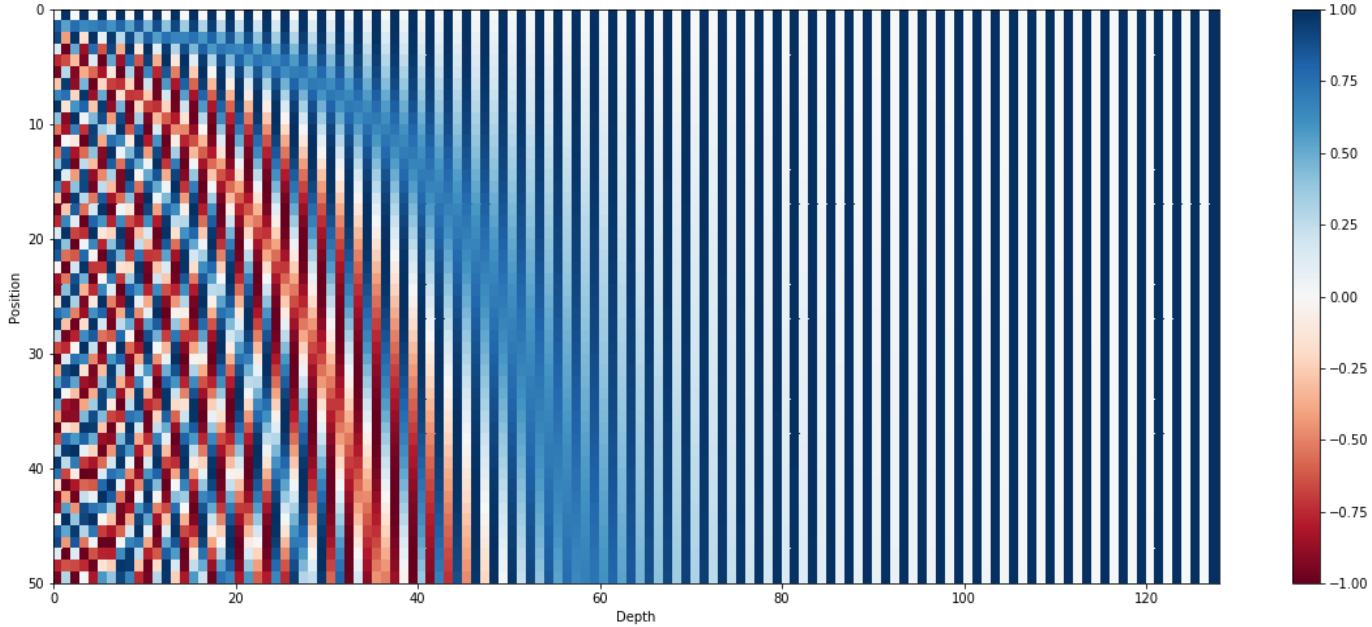
ith location in dim

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Vaswani et al. Attention is all you need. NeurIPS, 2017



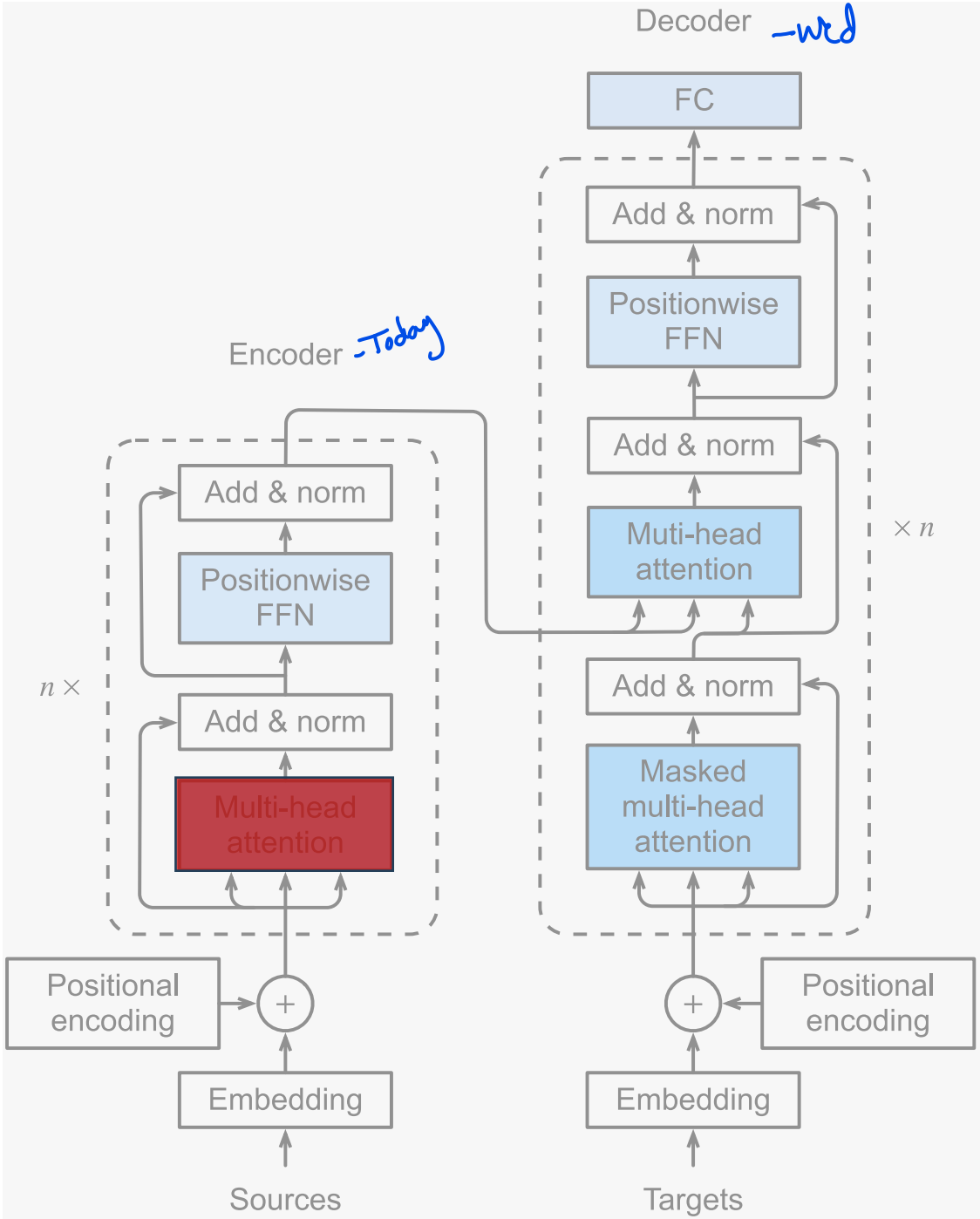
Frequencies for Positional Embedding



$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Vaswani et al. Attention is all you need. NeurIPS, 2017



Multi-Head Attention

But before that what is attention and why do we need it?

See video

Vaswani et al. Attention is all you need. NeurIPS, 2017

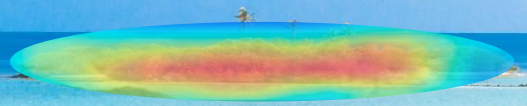
Psychology lessons: Selective Attention

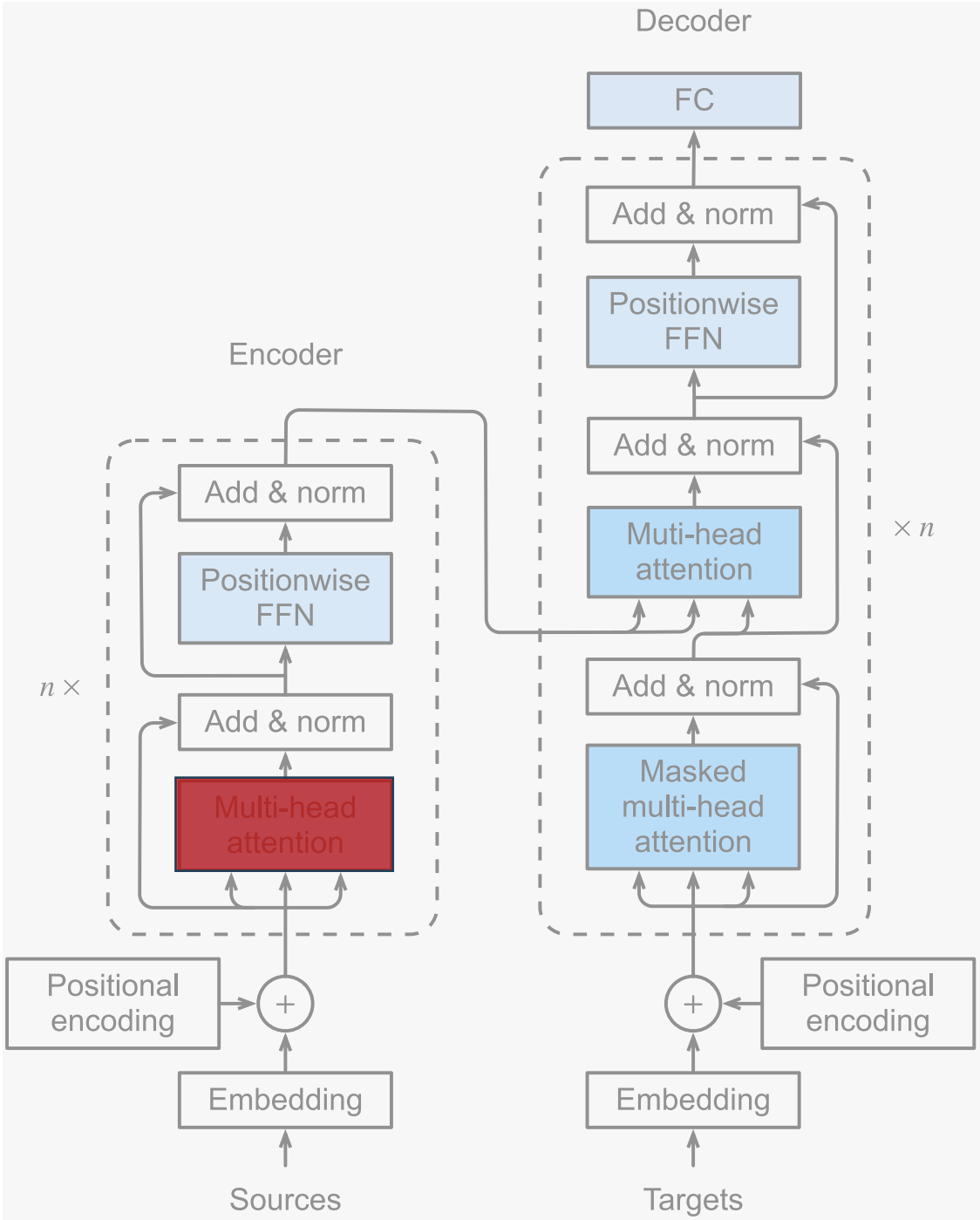
The act of focusing on a particular object for a period of time while simultaneously ignoring irrelevant information that is also occurring

Spotlight Effect!

We miss the gorilla running thru while counting, its the spotlight effect/ selective attention

we focus on the island first





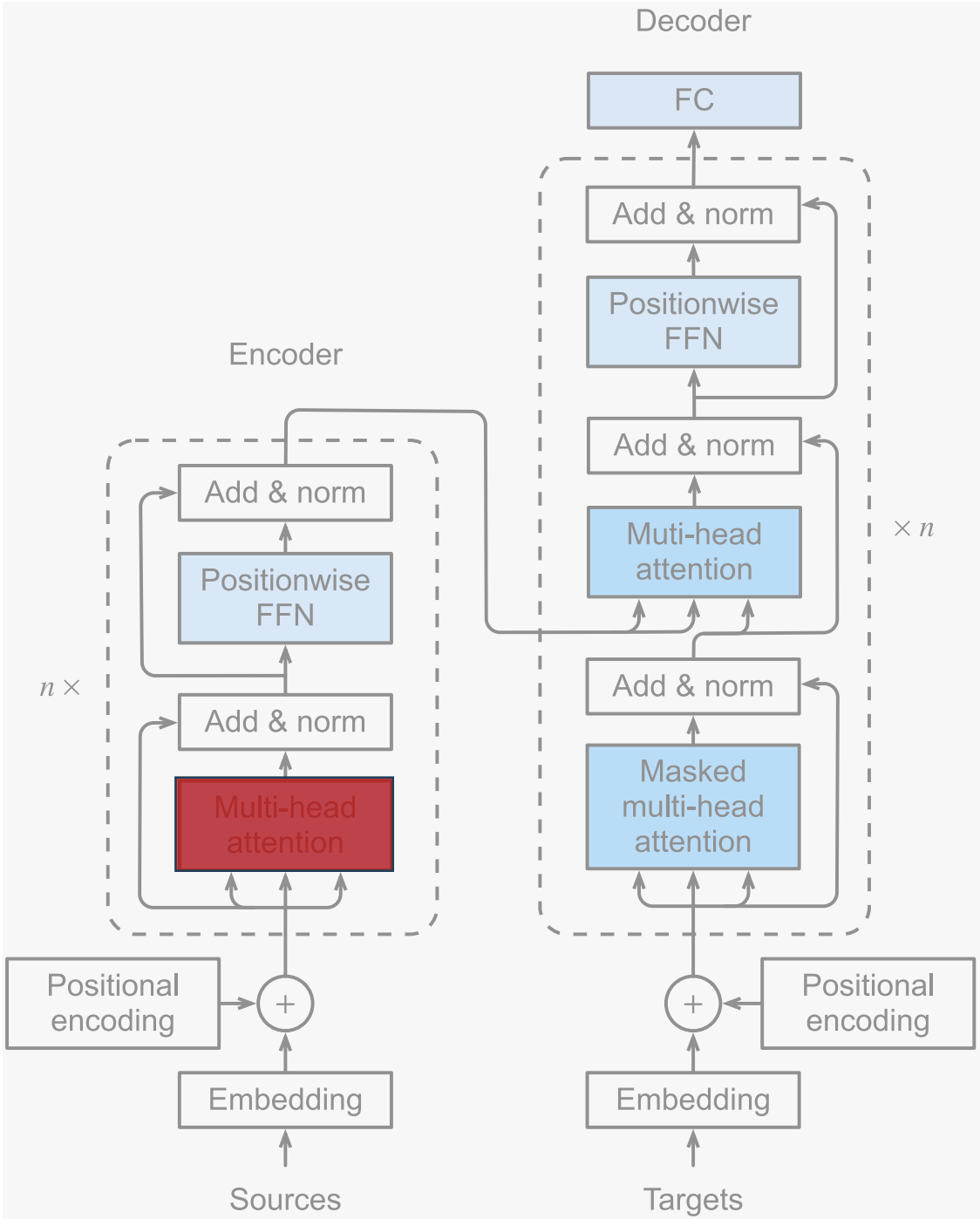
Attention

Who is this Harry Potter character?

Now if you two don't mind, I'm going to bed before either of you come up with another clever idea to get us killed - or worse, expelled.

Focus on important words in the sentence

Vaswani et al. Attention is all you need. NeurIPS, 2017

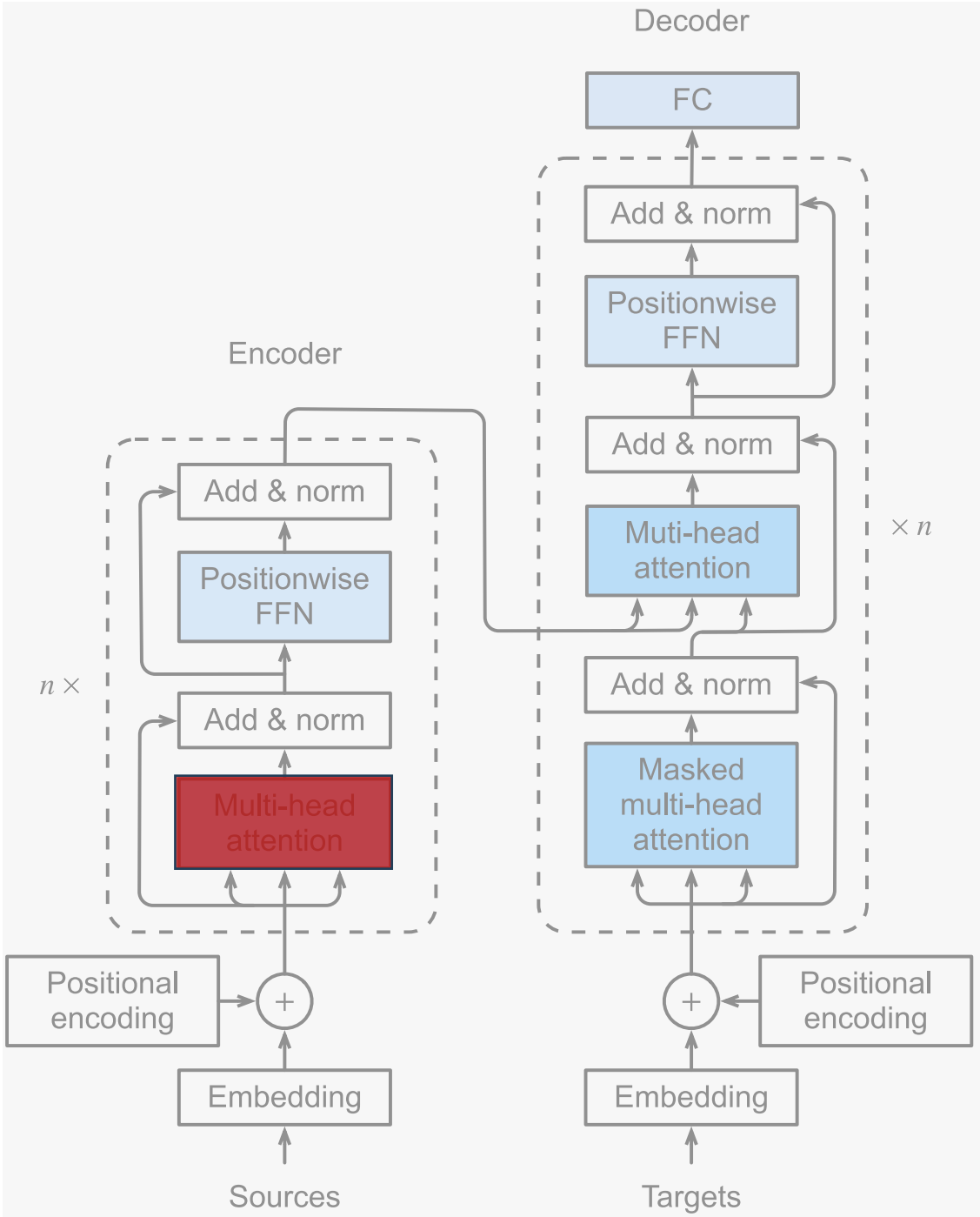


Attention

Who is this Harry Potter character?

Now if you two don't mind, I'm going to bed before either of you come up with another clever idea to get us killed - or worse, expelled.

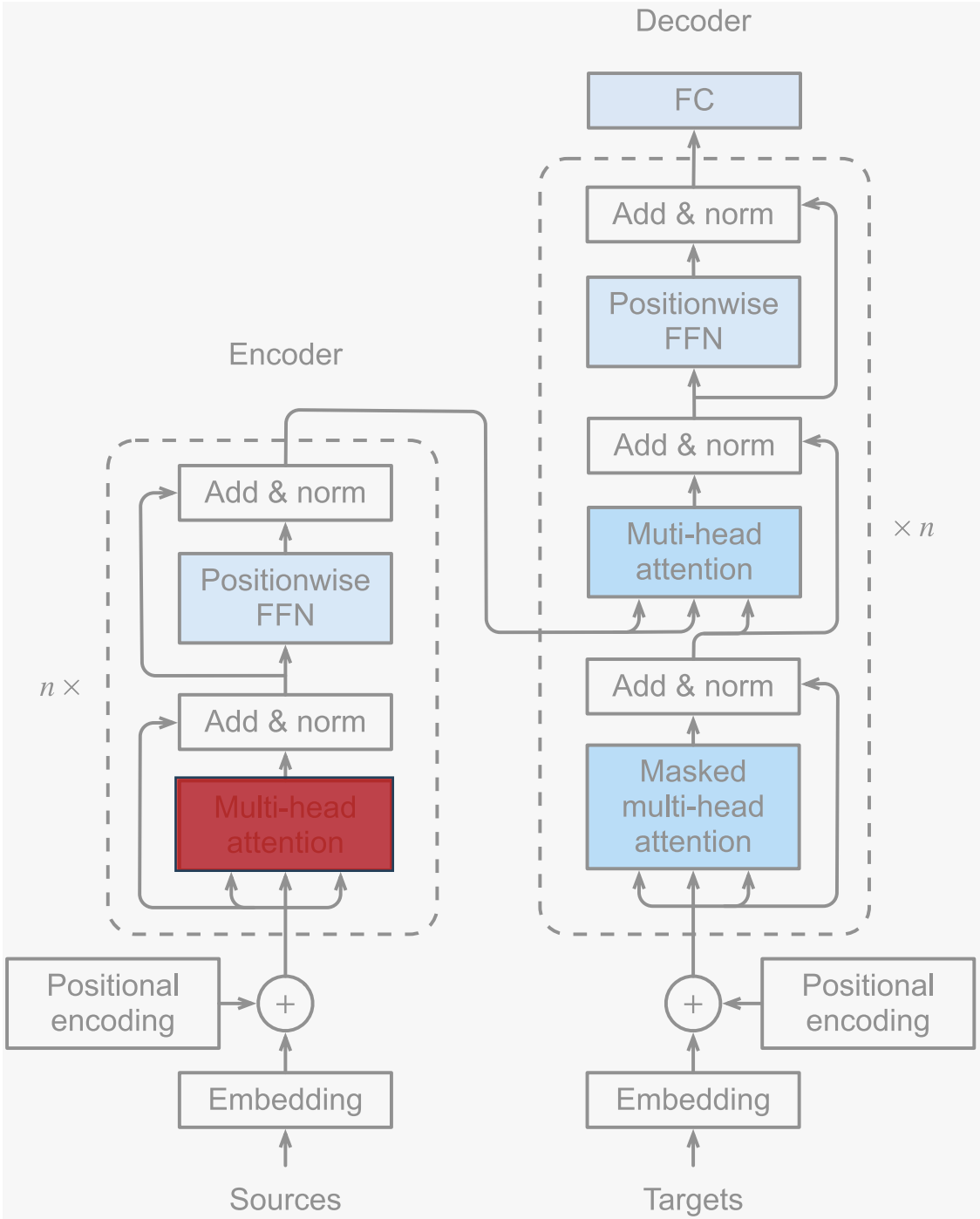
Vaswani et al. Attention is all you need. NeurIPS, 2017



Who is this Harry Potter character?

Now if you two don't mind, I'm going to bed before either of you come up with another clever idea to get us killed - or worse, expelled.

Vaswani et al. Attention is all you need. NeurIPS, 2017



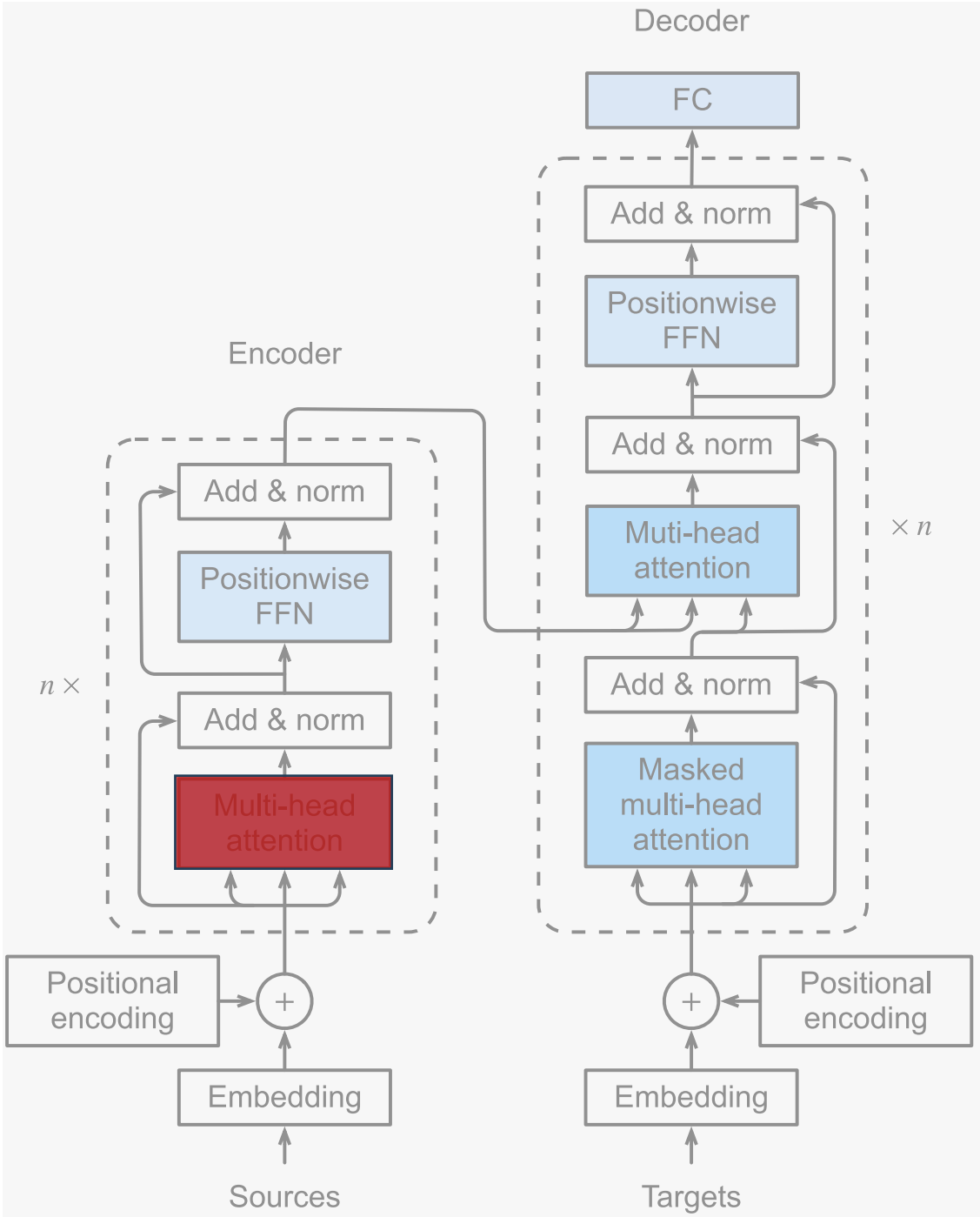
Self-Attention

vs cross attention (later)

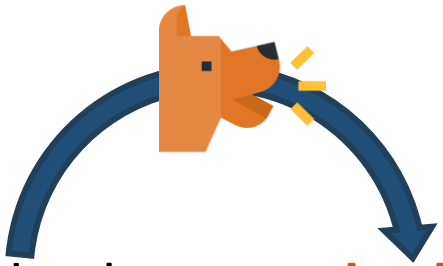
The dog began to **bark** loudly when it saw someone approaching the tree with rough **bark**.

Every context must be understood. bark vs bark, where dog is doing, etc

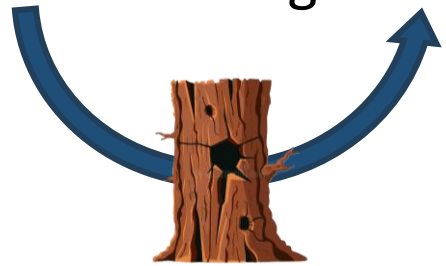
Vaswani et al. Attention is all you need. NeurIPS, 2017



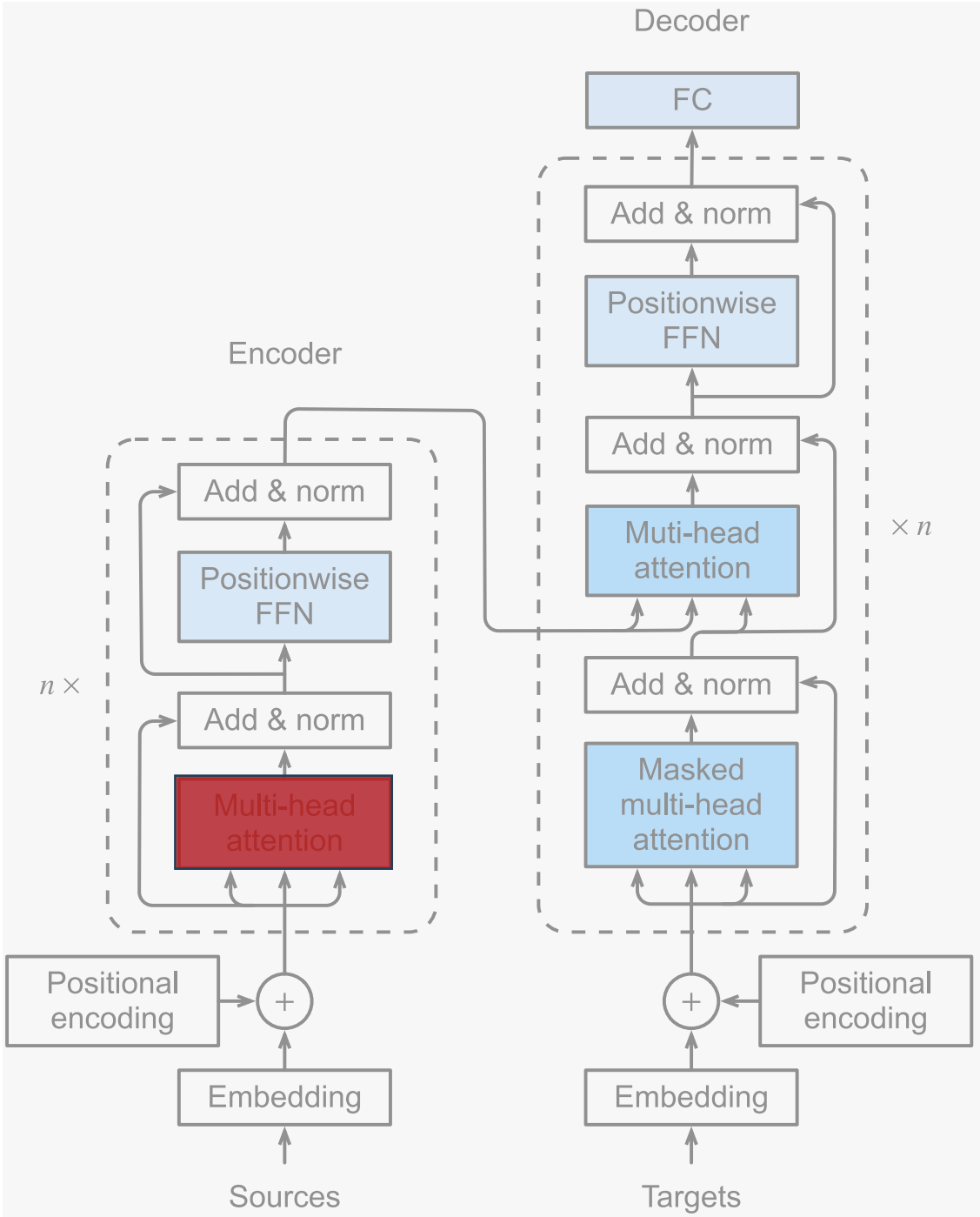
Self-Attention



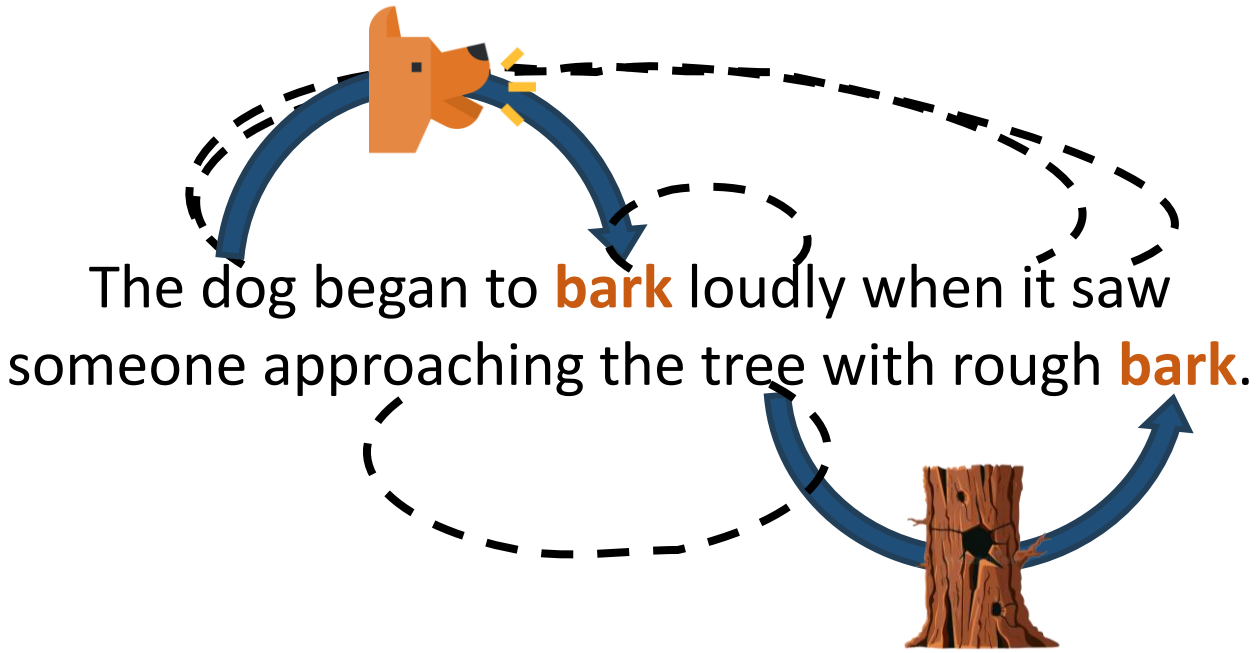
The dog began to **bark** loudly when it saw someone approaching the tree with rough **bark**.



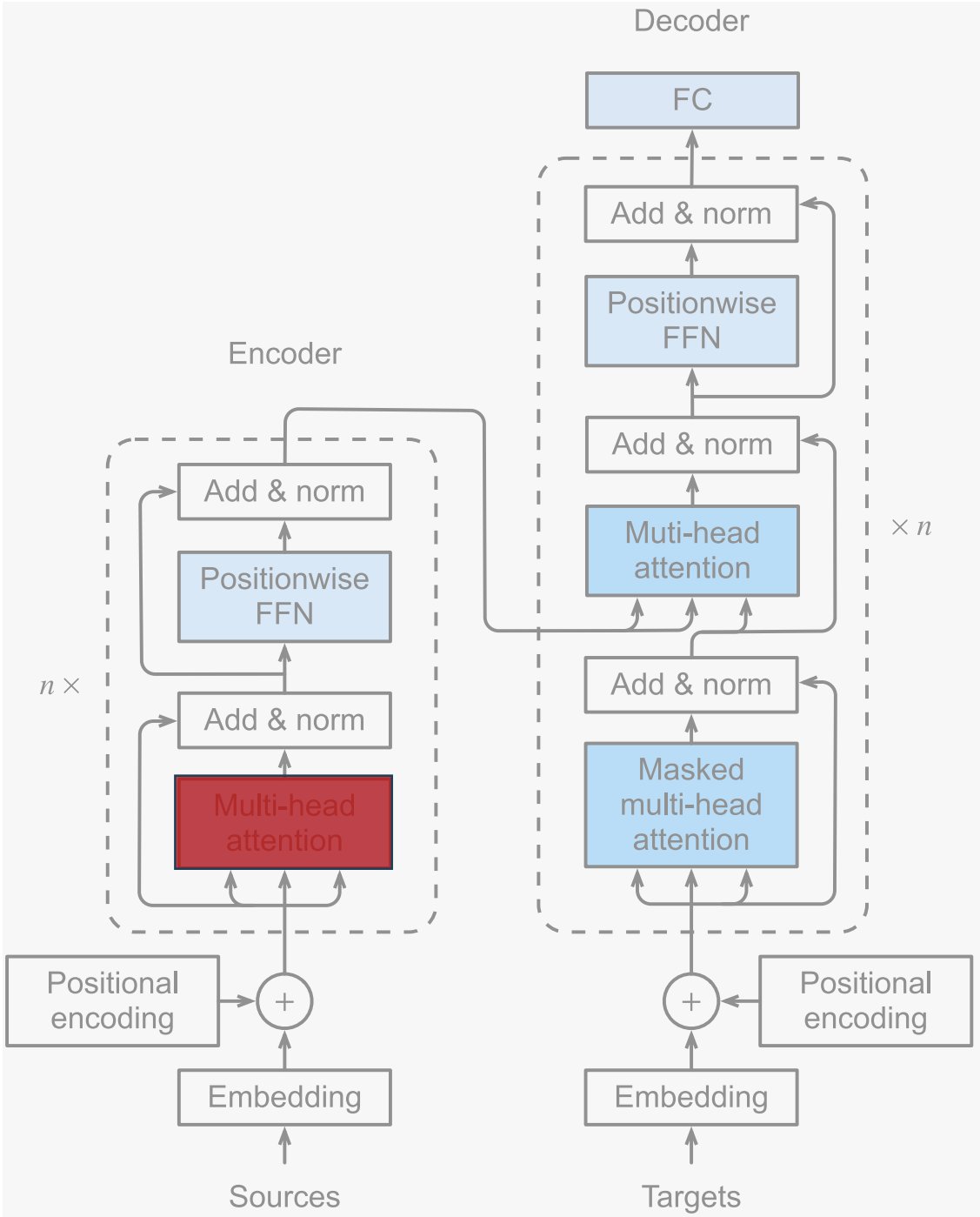
Vaswani et al. Attention is all you need. NeurIPS, 2017



Self-Attention



Vaswani et al. Attention is all you need. NeurIPS, 2017



Simple Attention

it attends to every word then the few needed to answer the question

Who is this Harry Potter character?

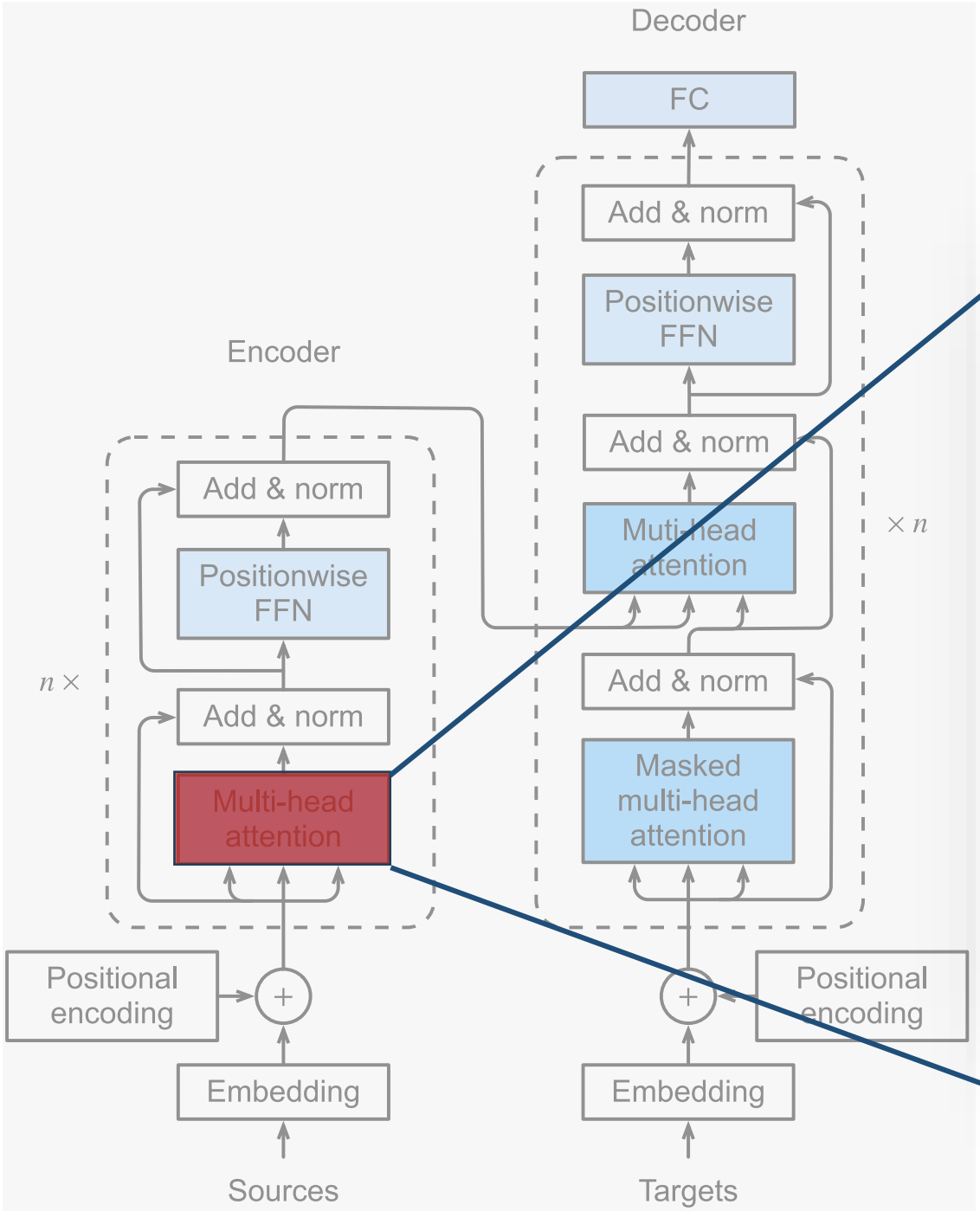
Now if you two don't mind, I'm going to bed before either of you come up with another clever idea to get us killed - or worse, expelled.

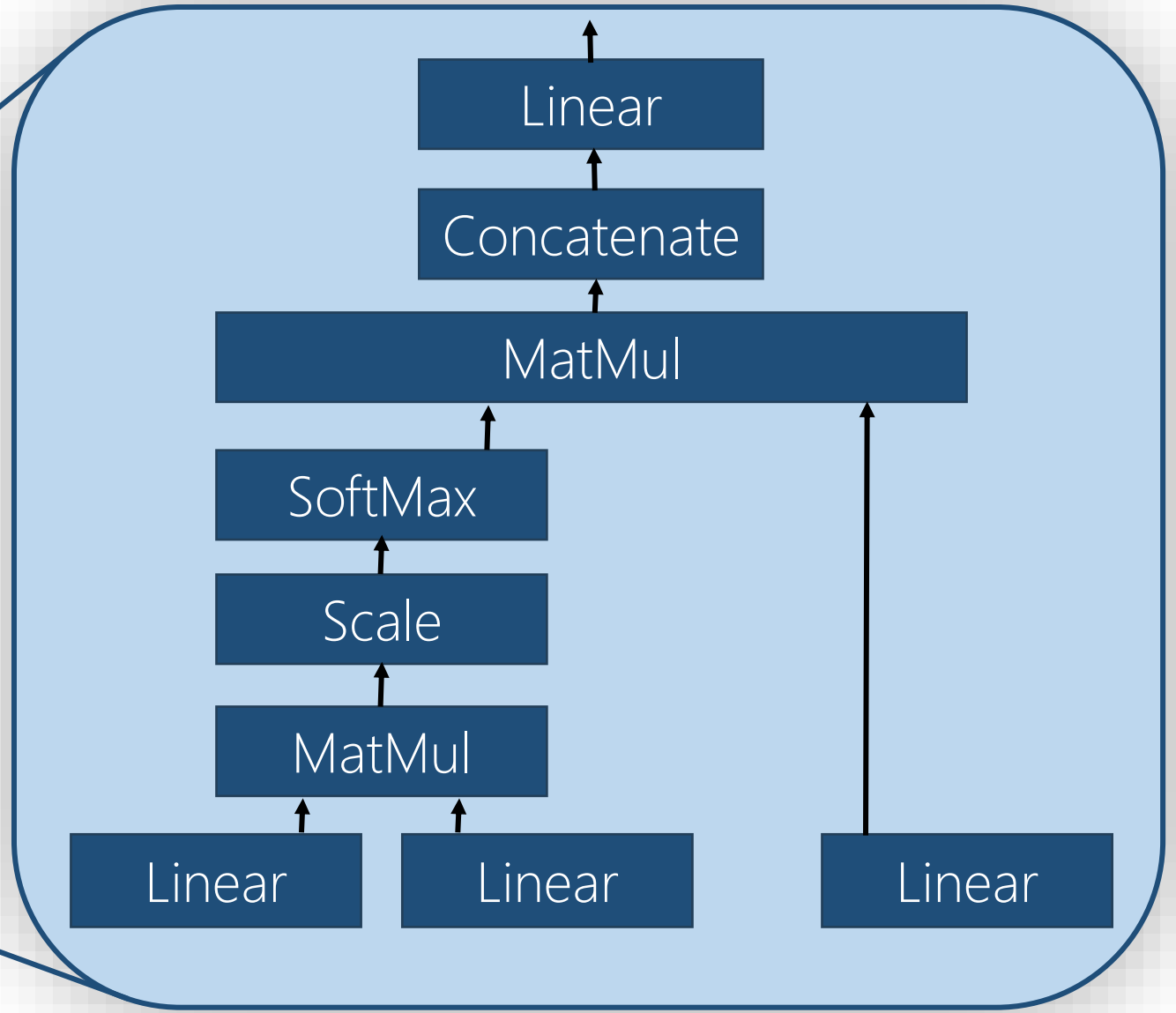
Self-Attention

The dog began to **bark** loudly when it saw someone approaching the tree with rough **bark**.

Vaswani et al. Attention is all you need. NeurIPS, 2017

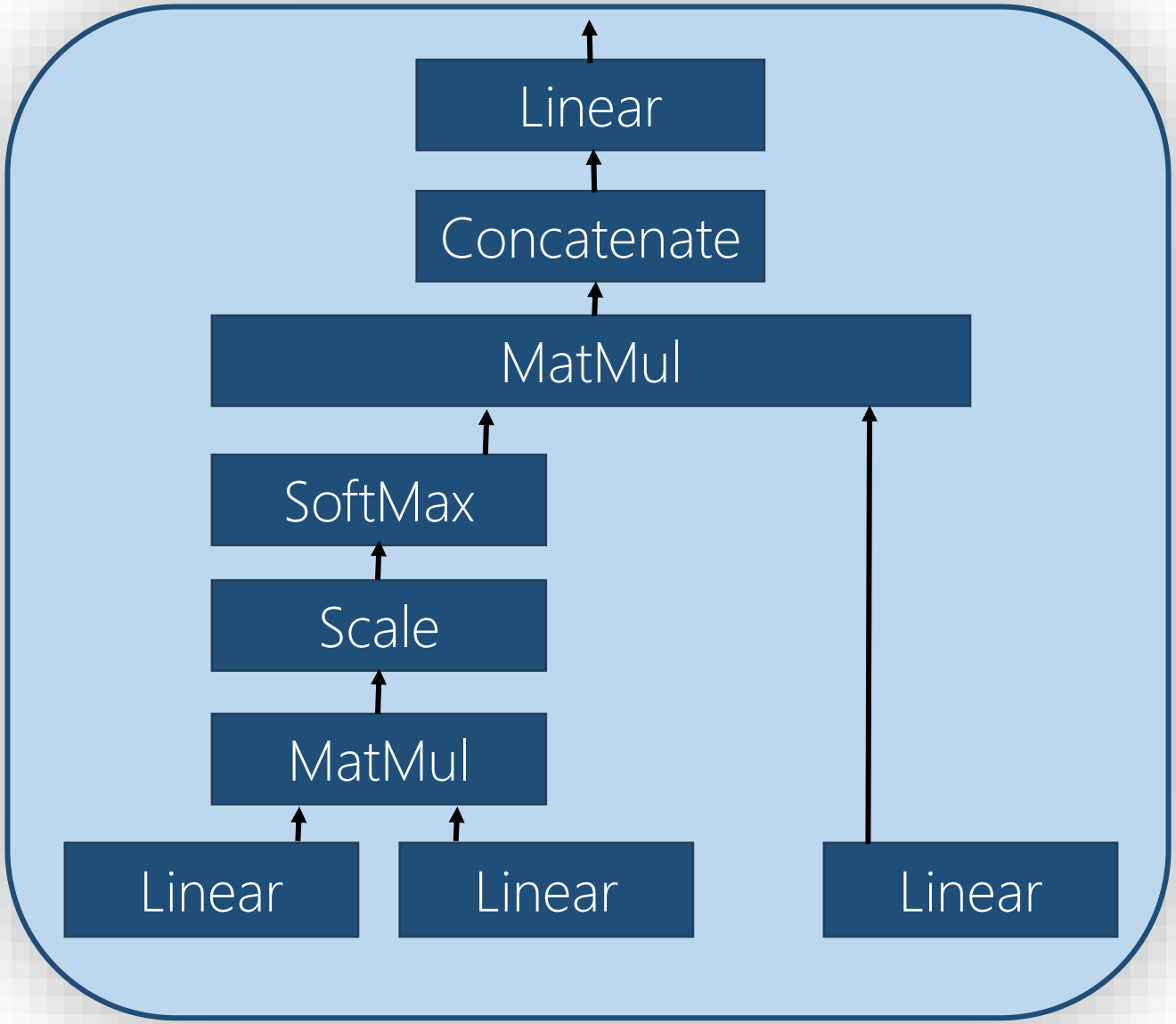
Multi-Head Attention





Vaswani et al. Attention is all you need. NeurIPS, 2017

Multi-Head Attention



linear transformation

concat diff vectors

dot product

normalizing to 0-1

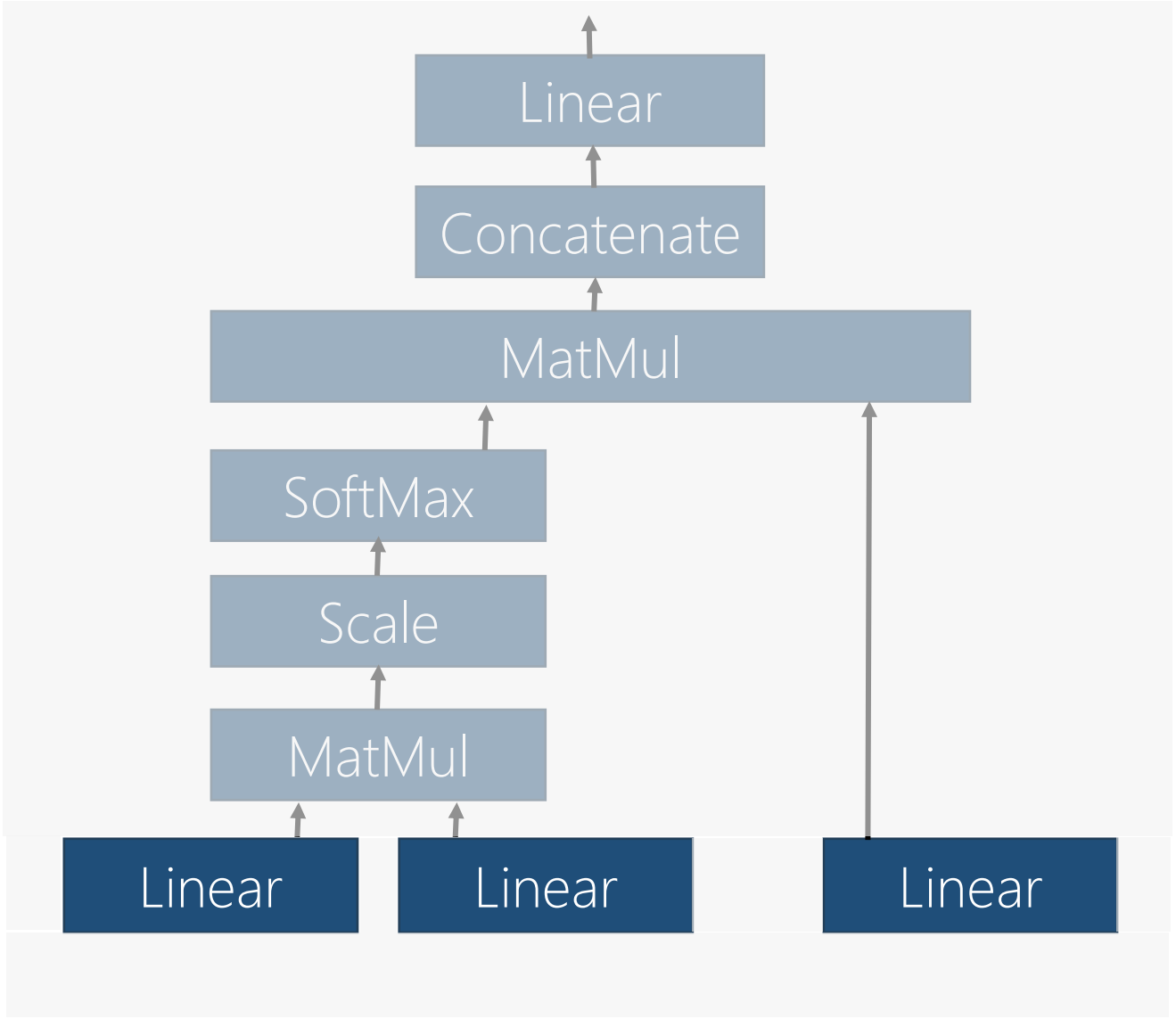
scaling

mat mul is the dot product

linear transform of the input

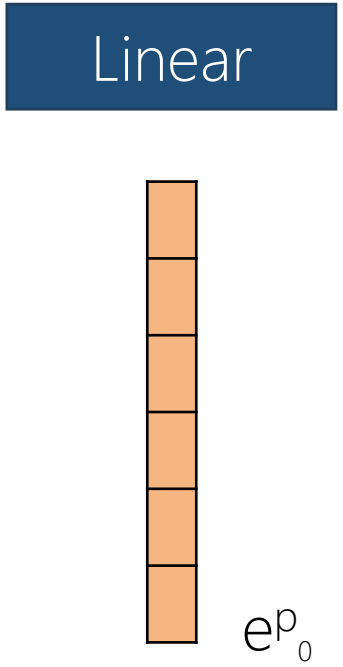
Vaswani et al. Attention is all you need. NeurIPS, 2017

Multi-Head Attention



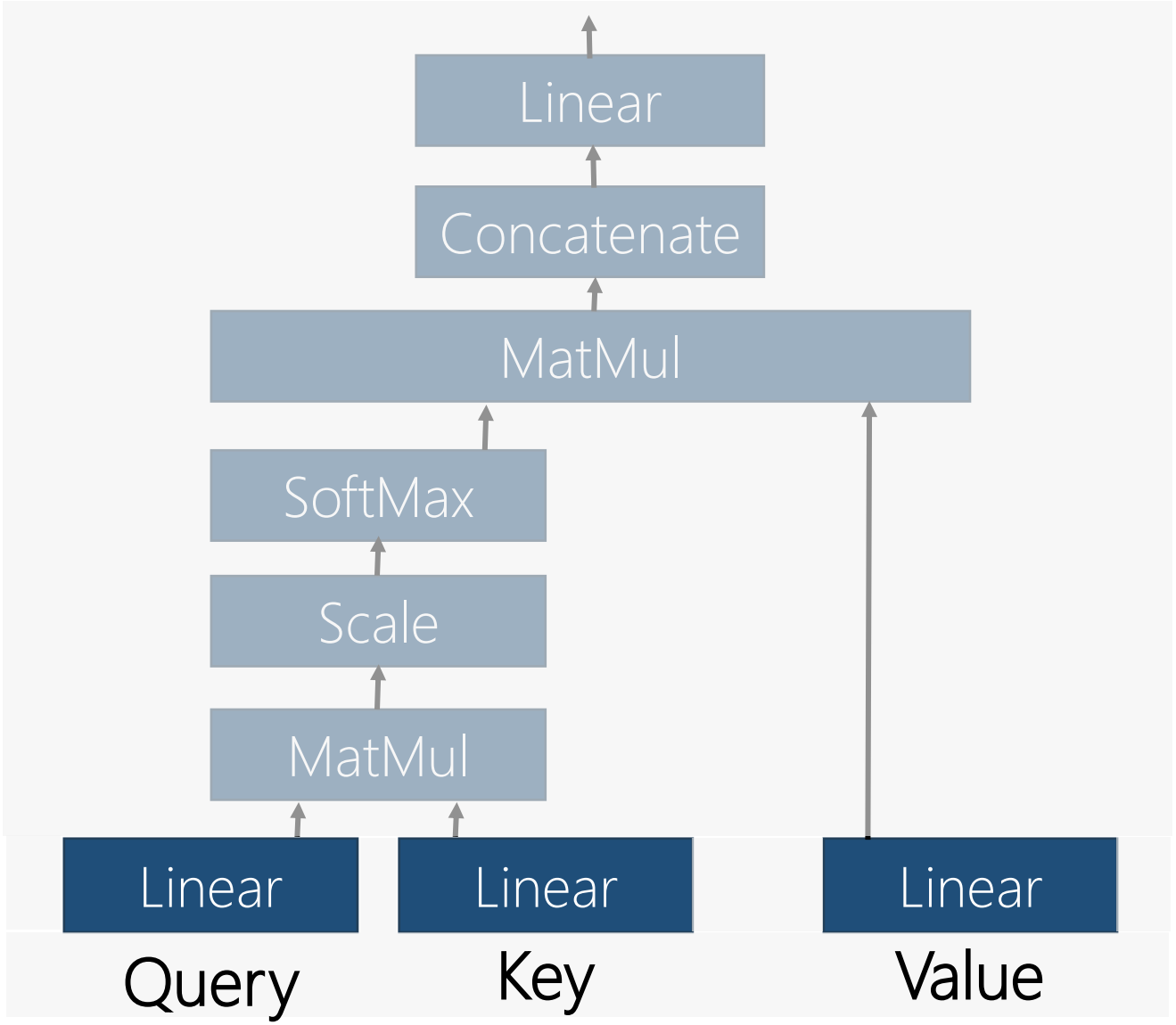
What does a linear layer do?

Input dims could be 128, 256 etc
It tries to breakdown to smaller dimension



Vaswani et al. Attention is all you need. NeurIPS, 2017

Multi-Head Attention



Analogous to Data Retrieval/Search

Query, key, value-foundational elements of attention

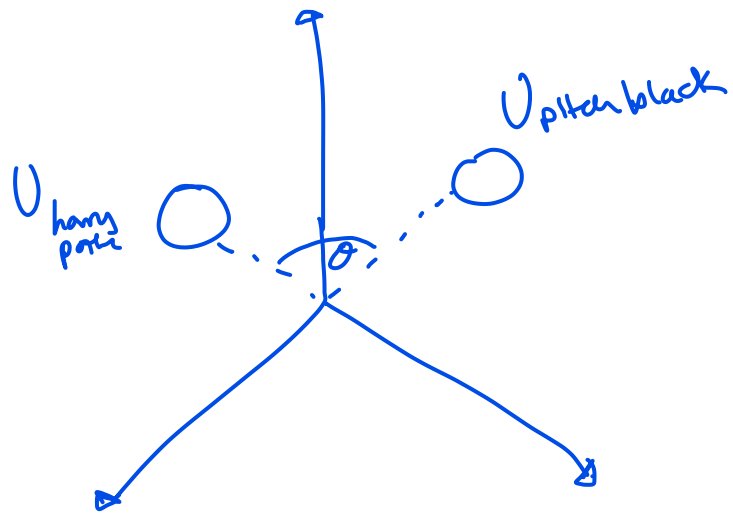
Query: info looking for
Key: where in database (of what the video is about in youtube for ex)
Value: what info can I share

inputting a 7x4 matrix for the query and key: 7= # words, 4 is the dim of the words

Vaswani et al. Attention is all you need. NeurIPS, 2017

Similarity between Query and Key?

input the image I took



- $\cos(\theta) = 1$

  $\theta = 0 \rightarrow$ A and B point in same direction

- $\cos(\theta) = -1$

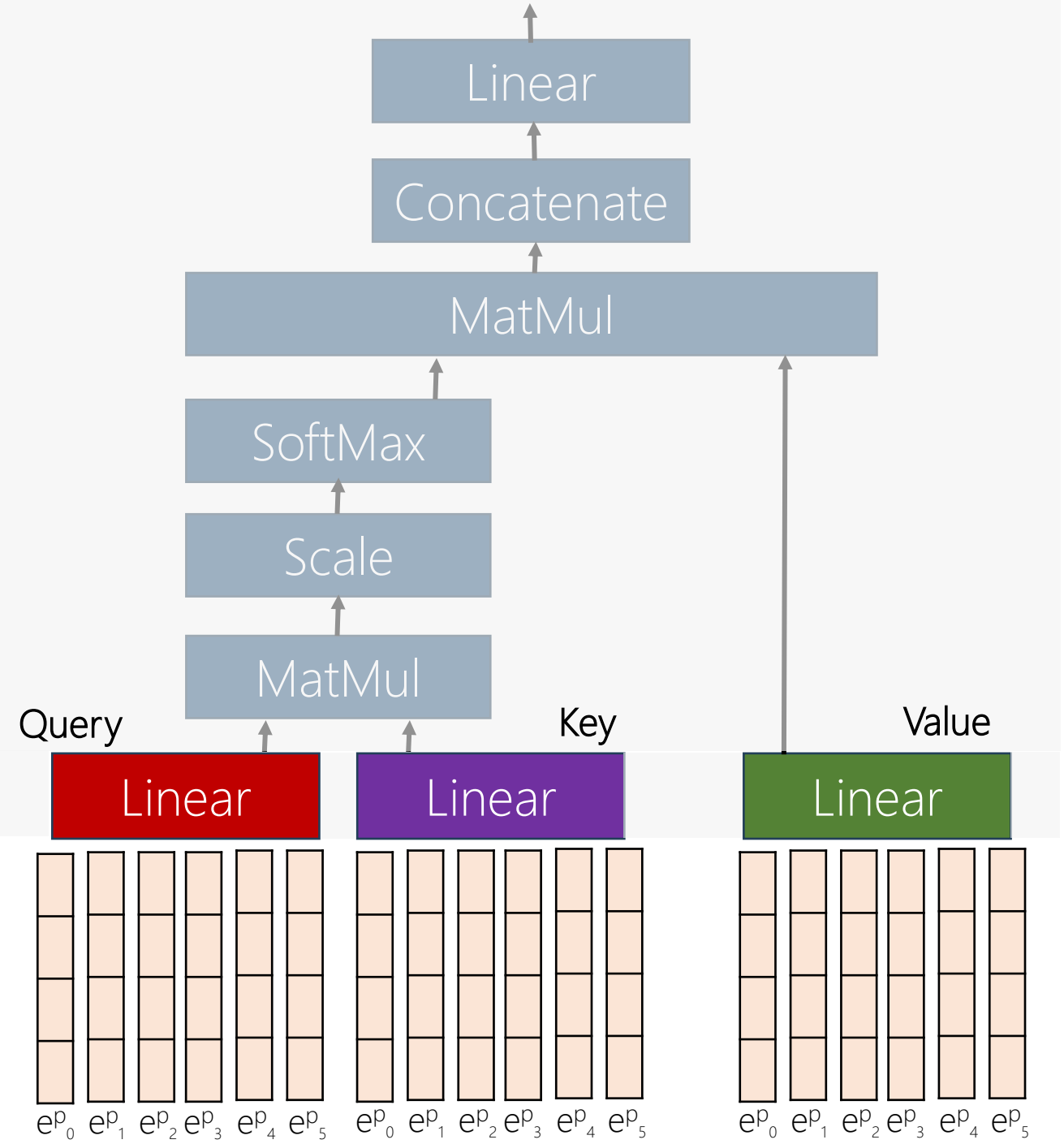
  $\theta = 180 \rightarrow$ A and B pt in opp directions

- $\cos(\theta) = 0$

  $\theta = 90 \rightarrow$ A and B @ 90° (orthogonal)

Vaswani et al. Attention is all you need. NeurIPS, 2017

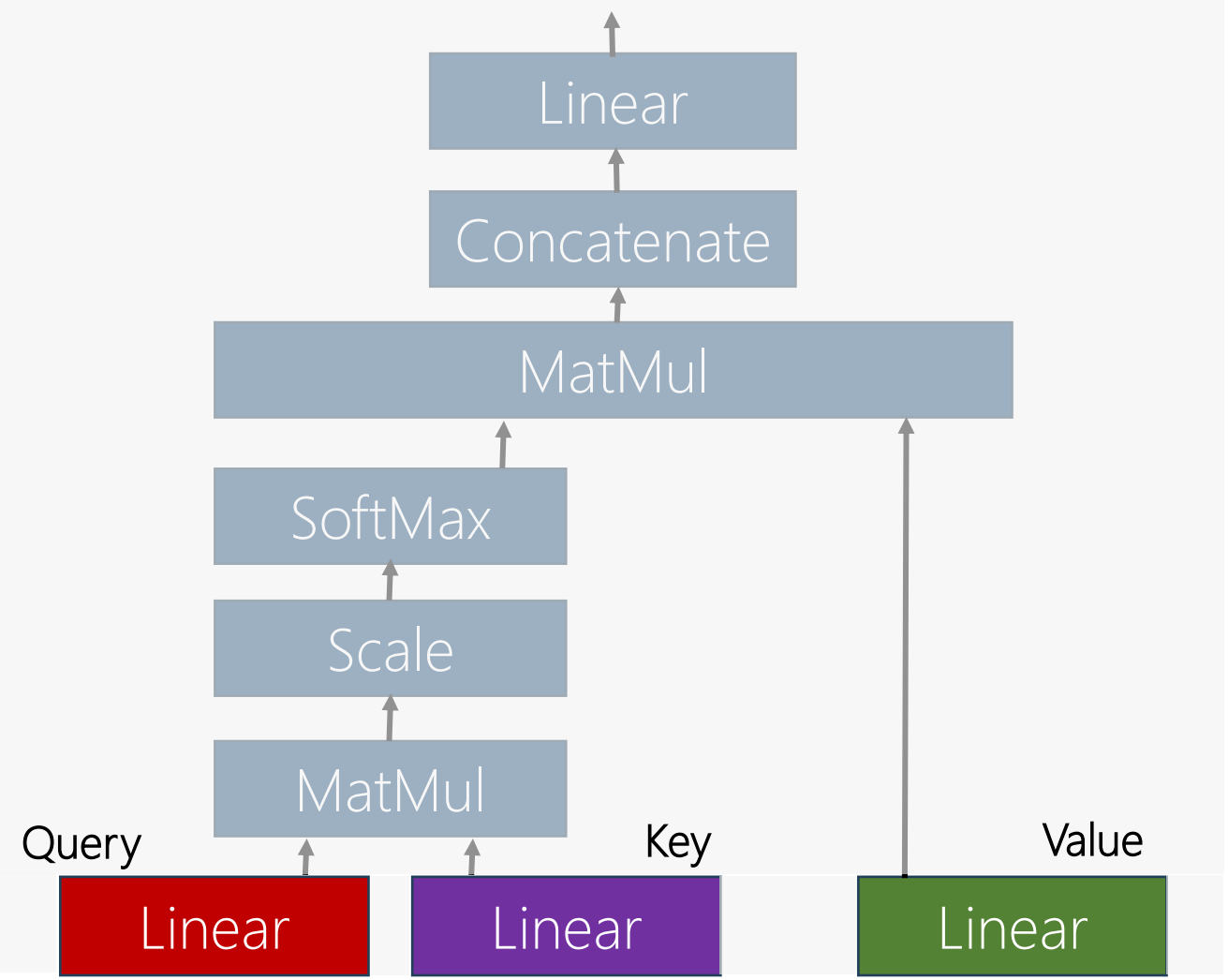
Multi-Head Attention

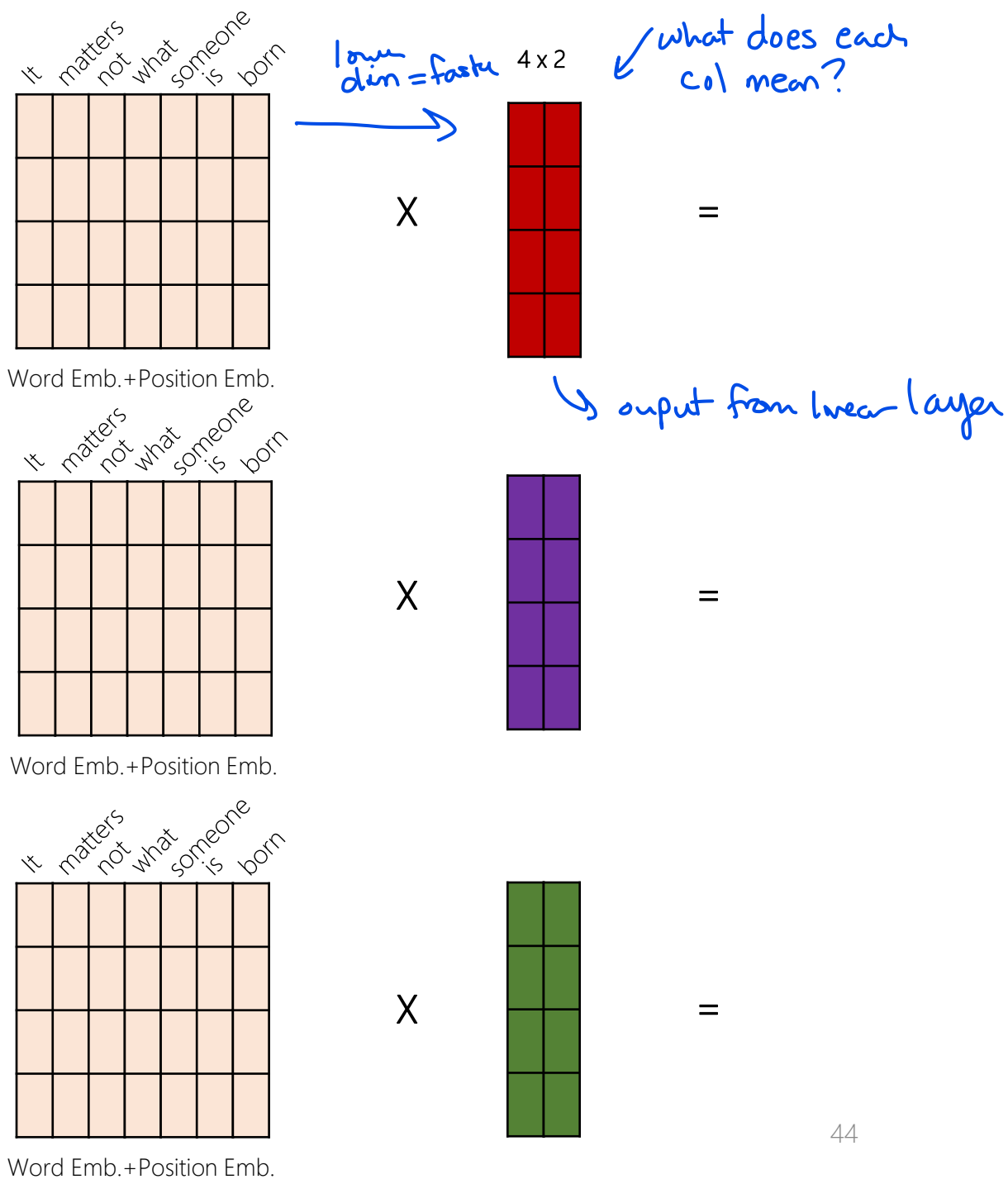


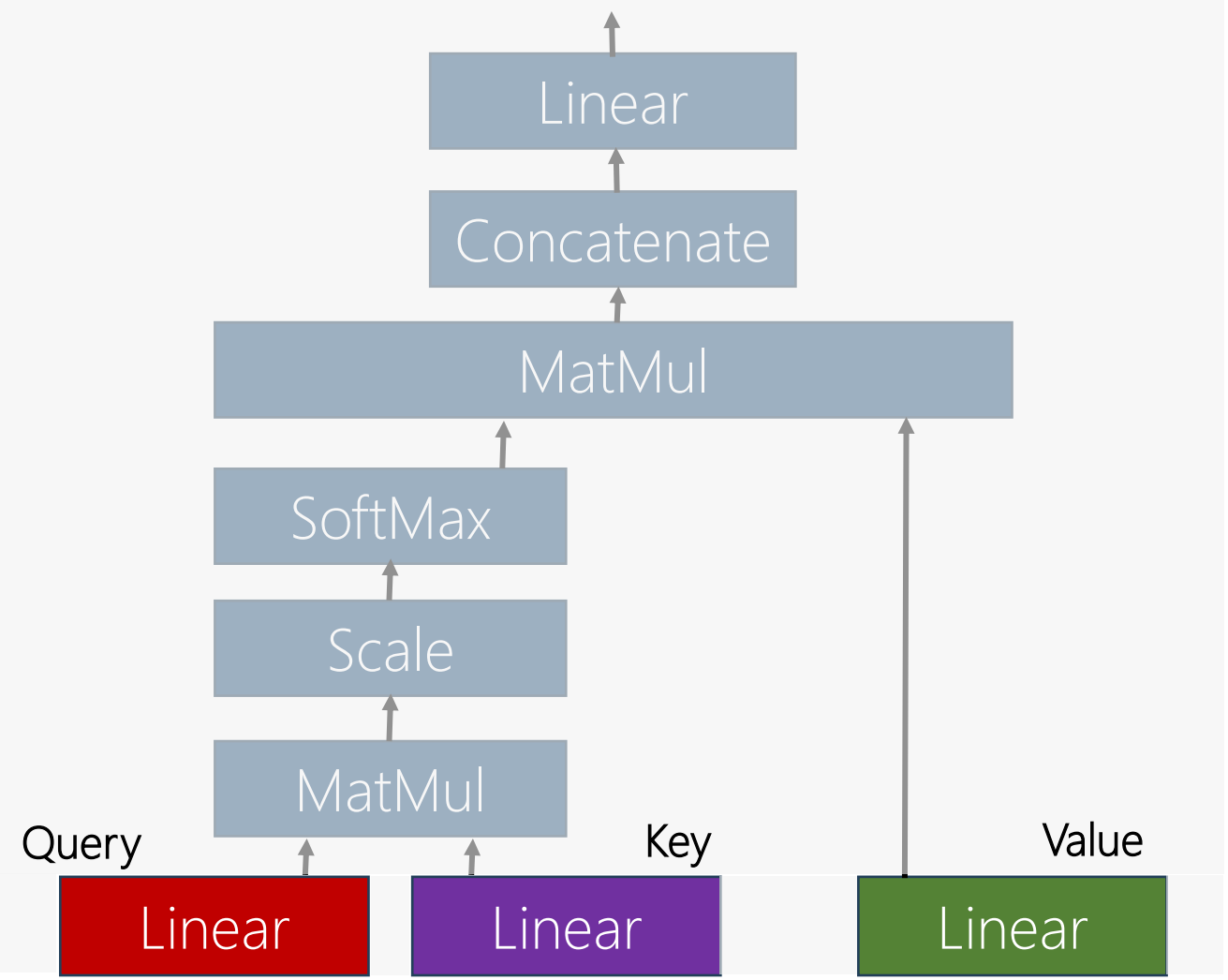
Analogous to Data Retrieval/Search

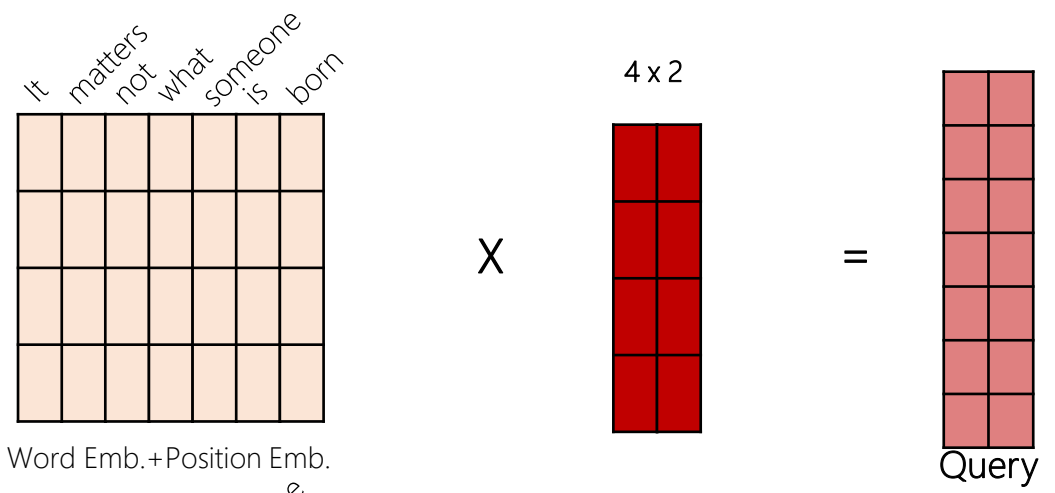
Not sending one word @ a time

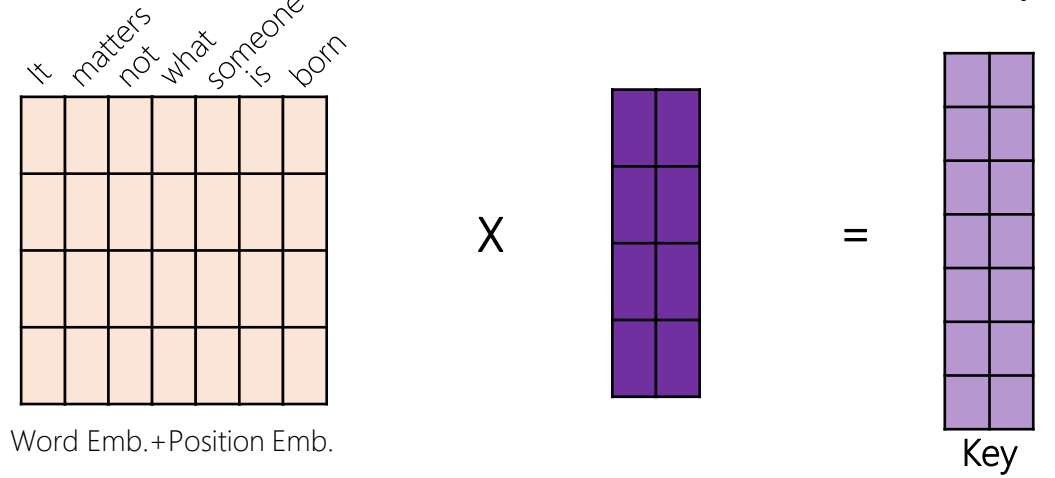
Vaswani et al. Attention is all you need. NeurIPS, 2017

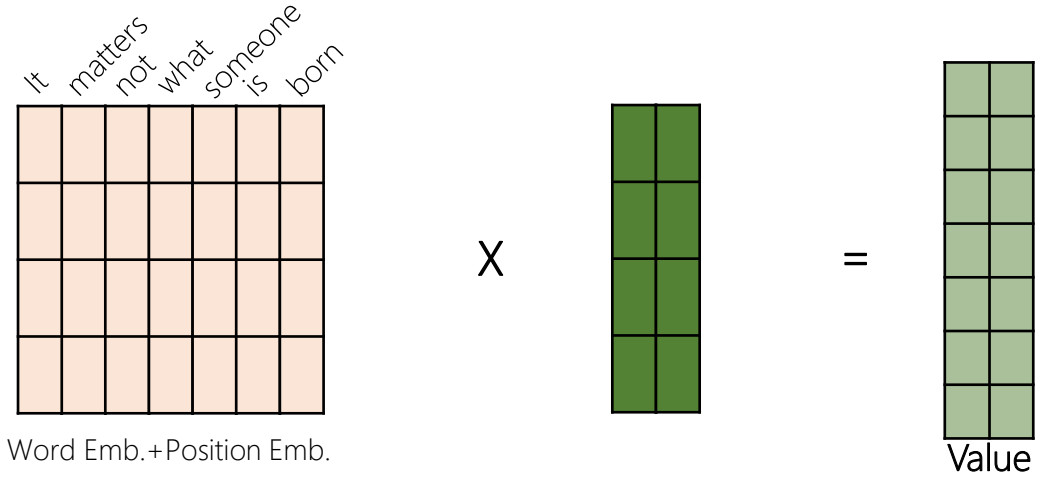

Linear
Concatenate
MatMul
SoftMax
Scale
MatMul
Query
Key
Value
Linear
Linear
Linear

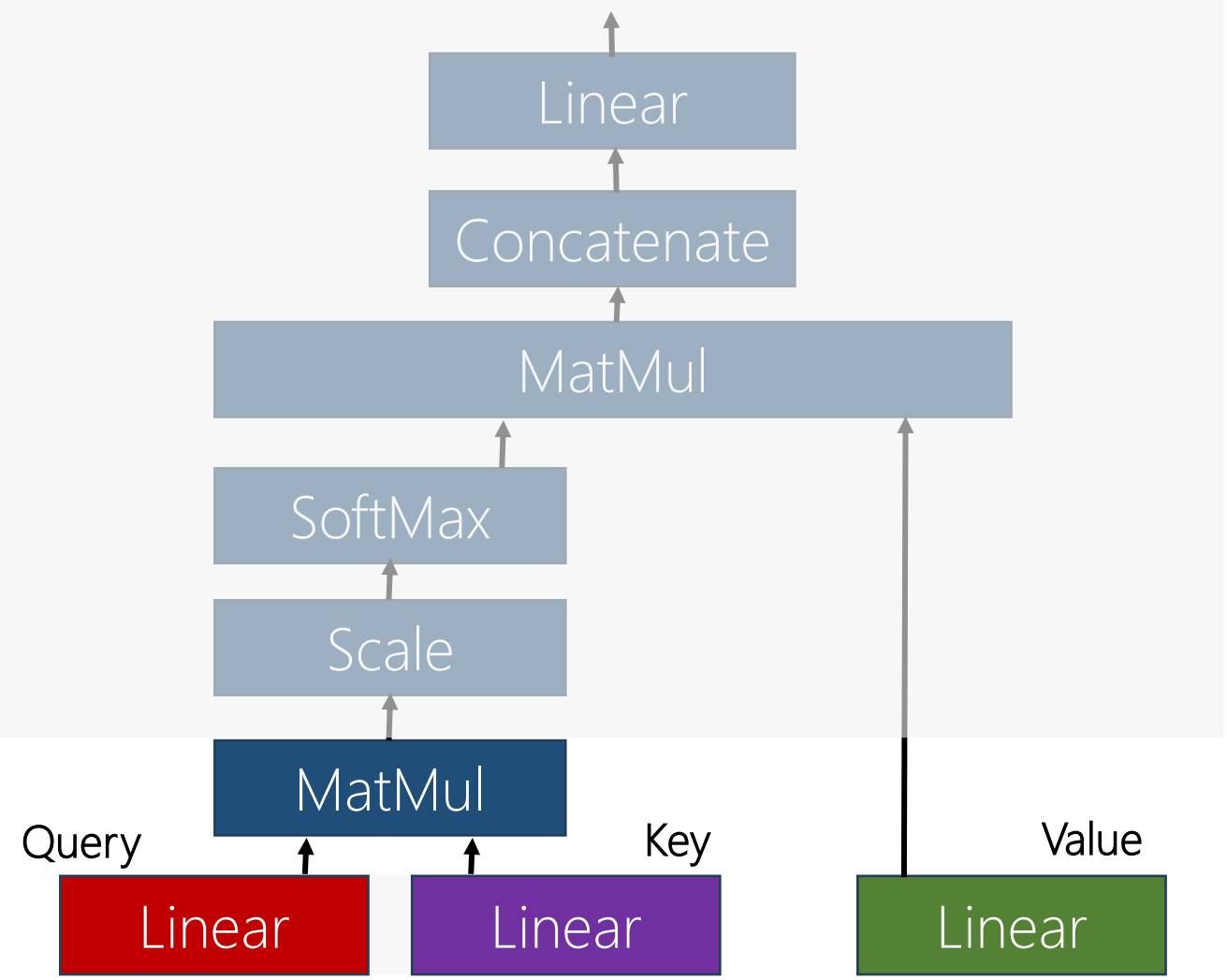

It matters not what someone is born
lower dim = faster
4 x 2
what does each col mean?
X
=
Word Emb.+Position Emb.
↳ output from linear layer
It matters not what someone is born
X
=
Word Emb.+Position Emb.
It matters not what someone is born
X
=
Word Emb.+Position Emb.


Linear
Concatenate
MatMul
SoftMax
Scale
MatMul
Query
Key
Value
Linear
Linear
Linear

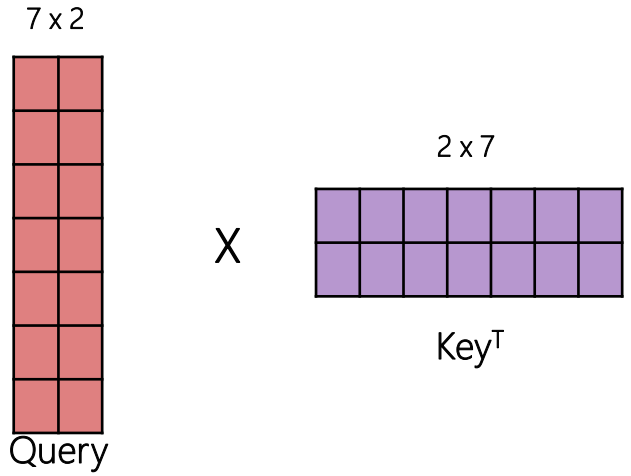

It matters not what someone is born
Word Emb.+Position Emb.
4 x 2
X
=
Query

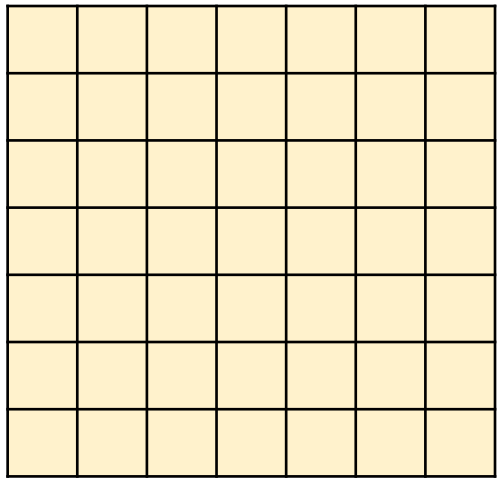

It matters not what someone is born
Word Emb.+Position Emb.
X
=
Key


It matters not what someone is born
Word Emb.+Position Emb.
X
=
Value

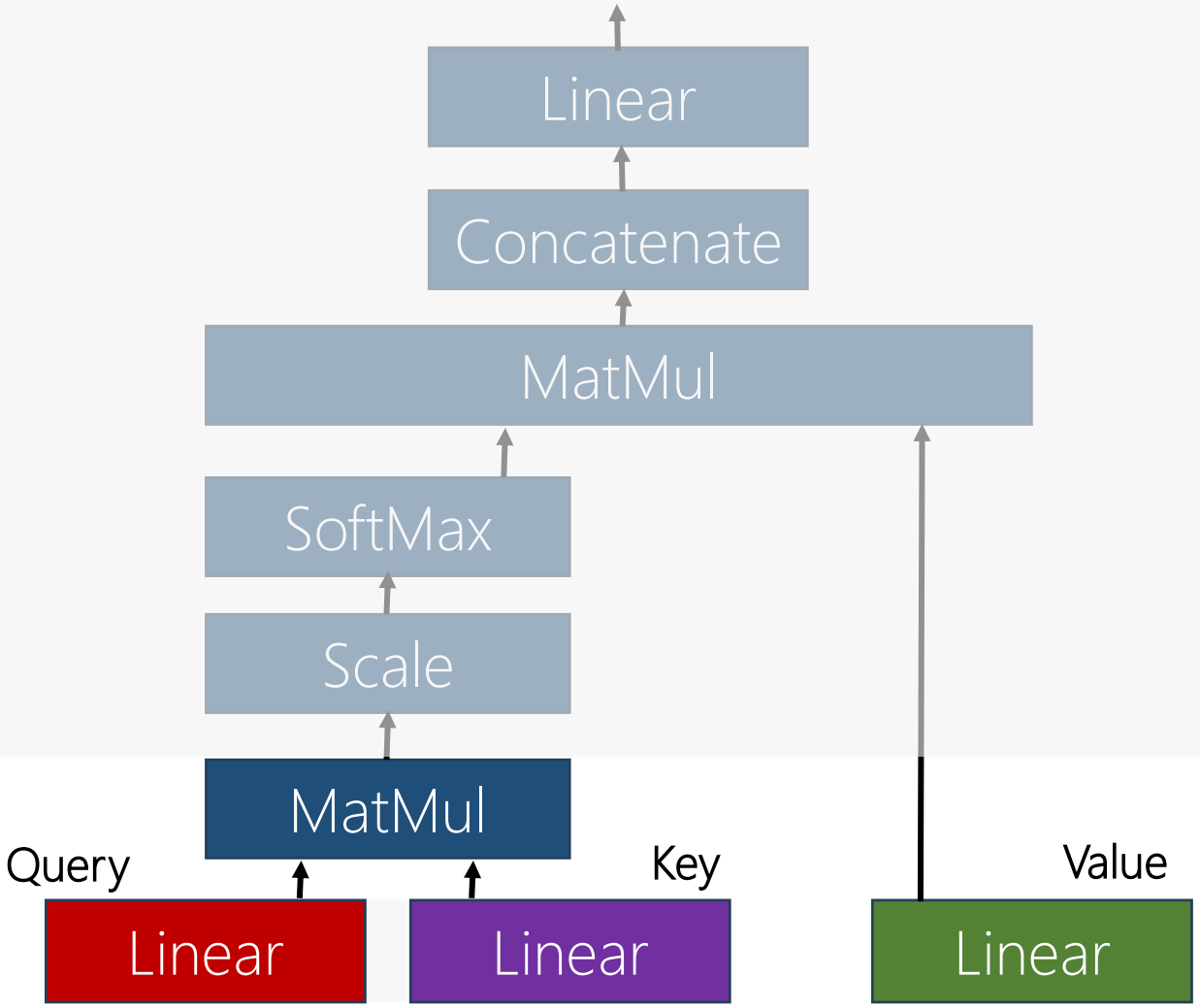


Attention Filter

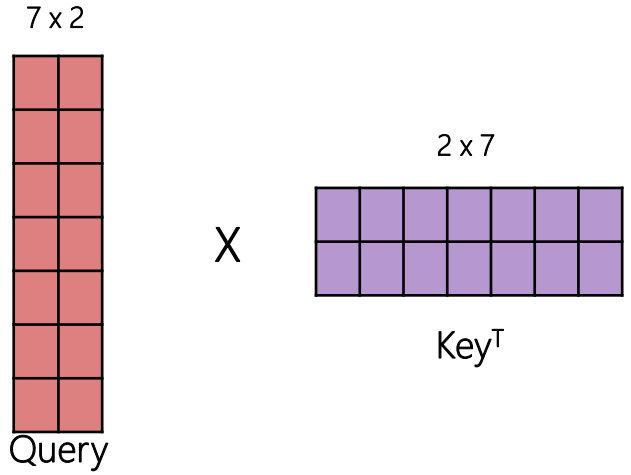




self attention matrix

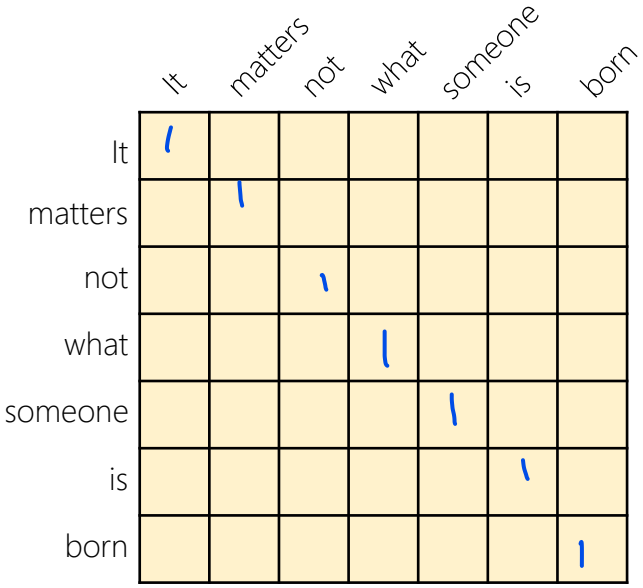


Attention Filter



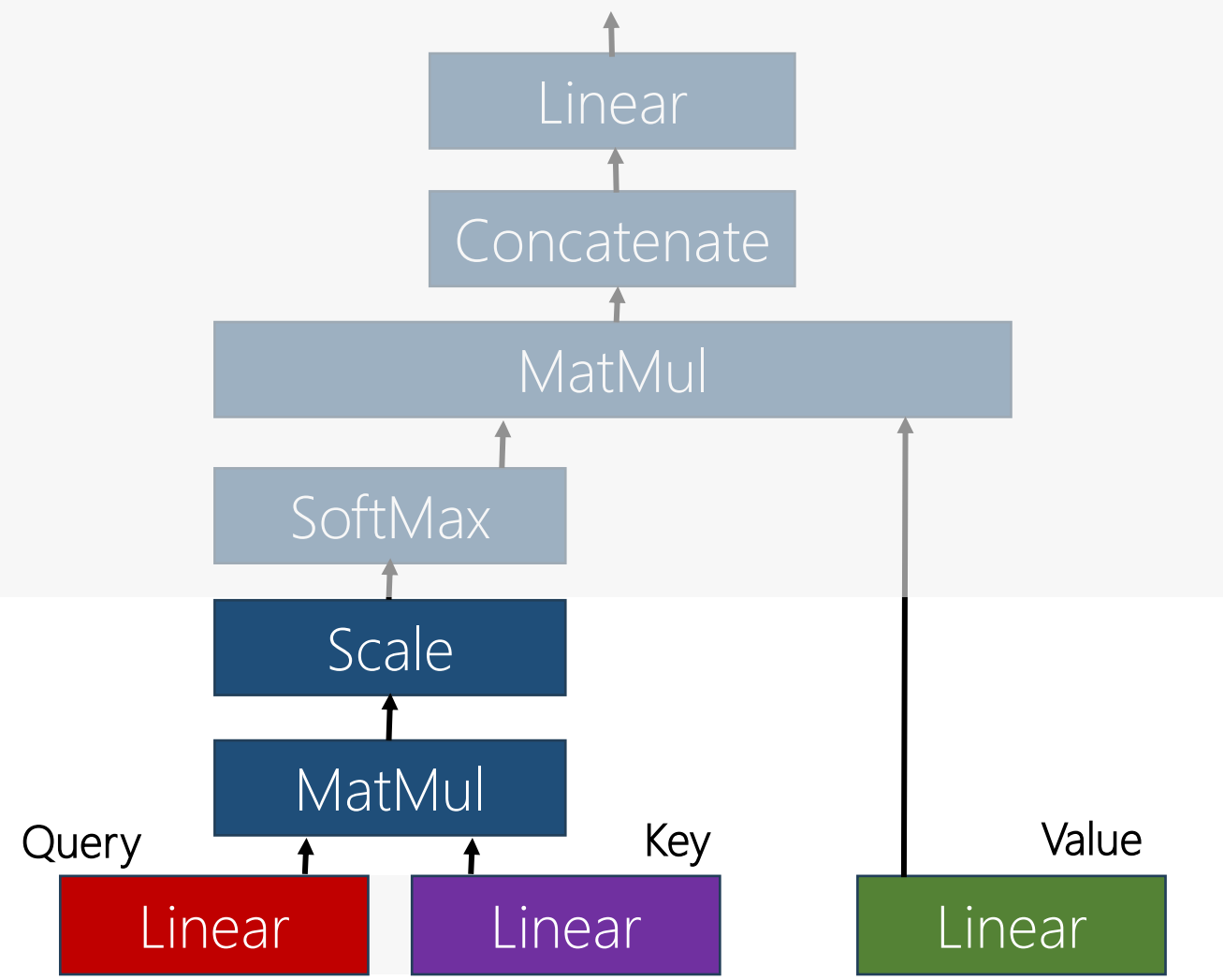
self attention matrix

Q: What do the vals rep?

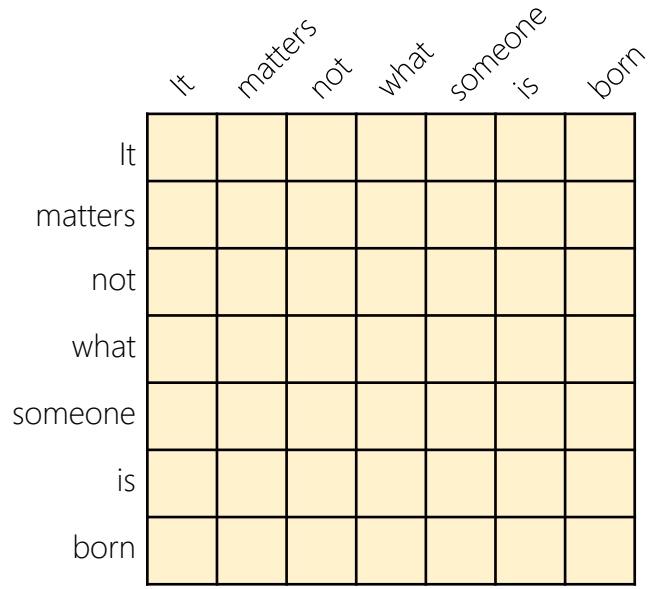


← Not banded nums which populate

← diag = 1

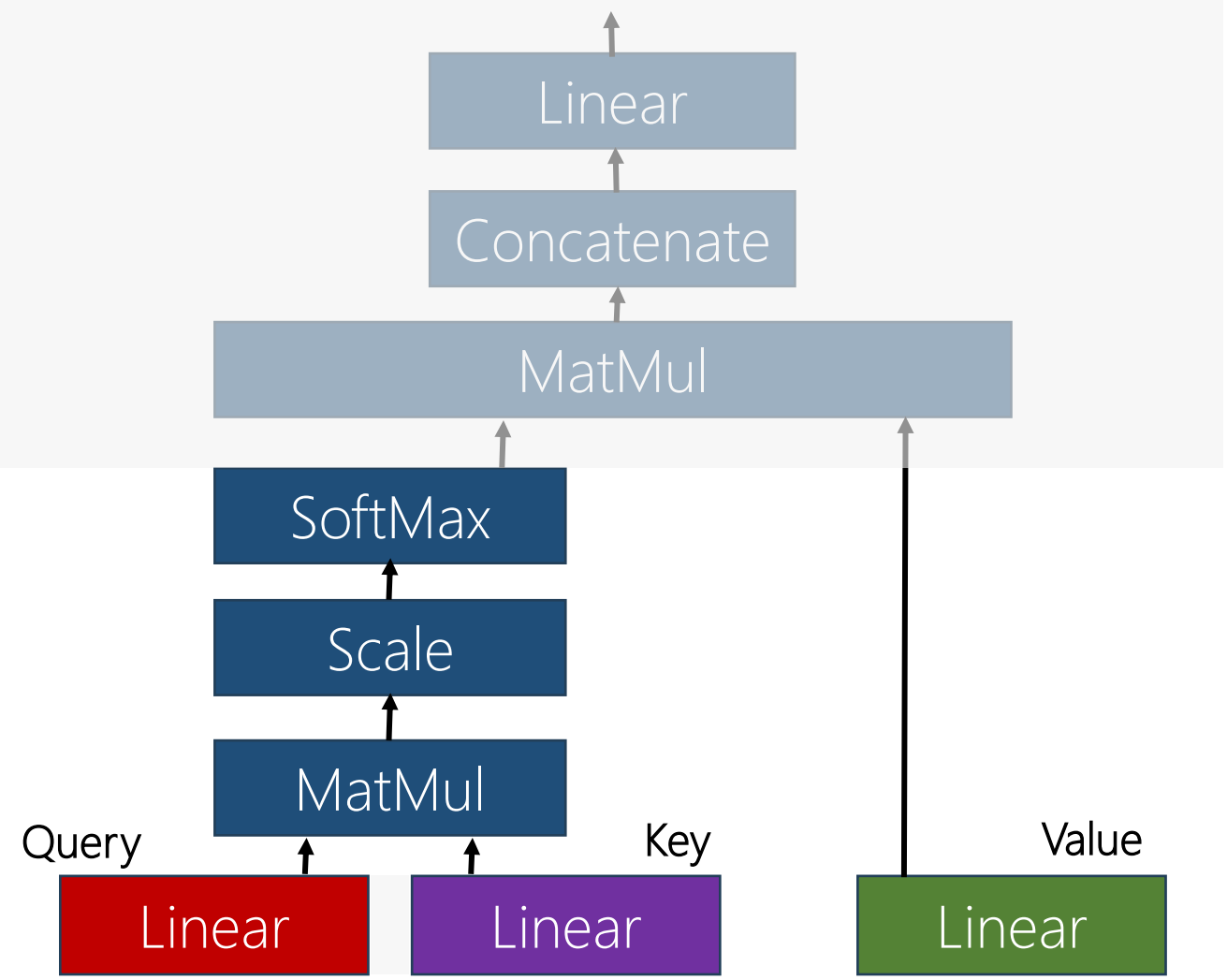


Attention Filter

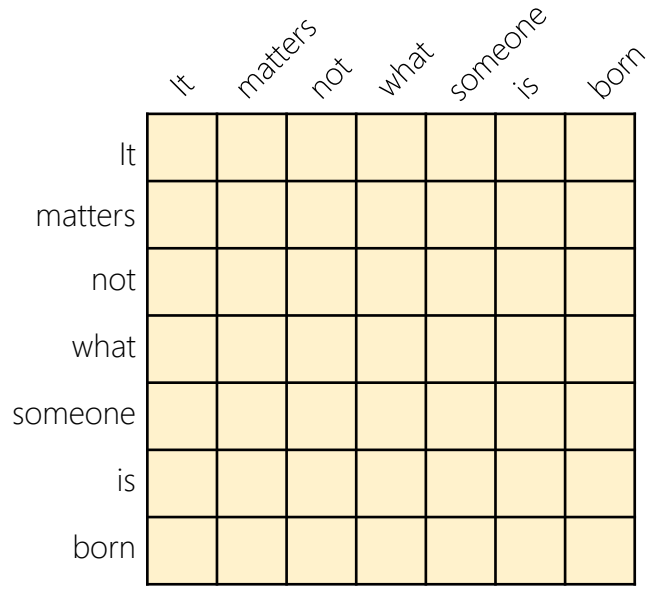


Scale using dimension! of the vector

(Still not in a closed space of 0 → 1



Attention Filter

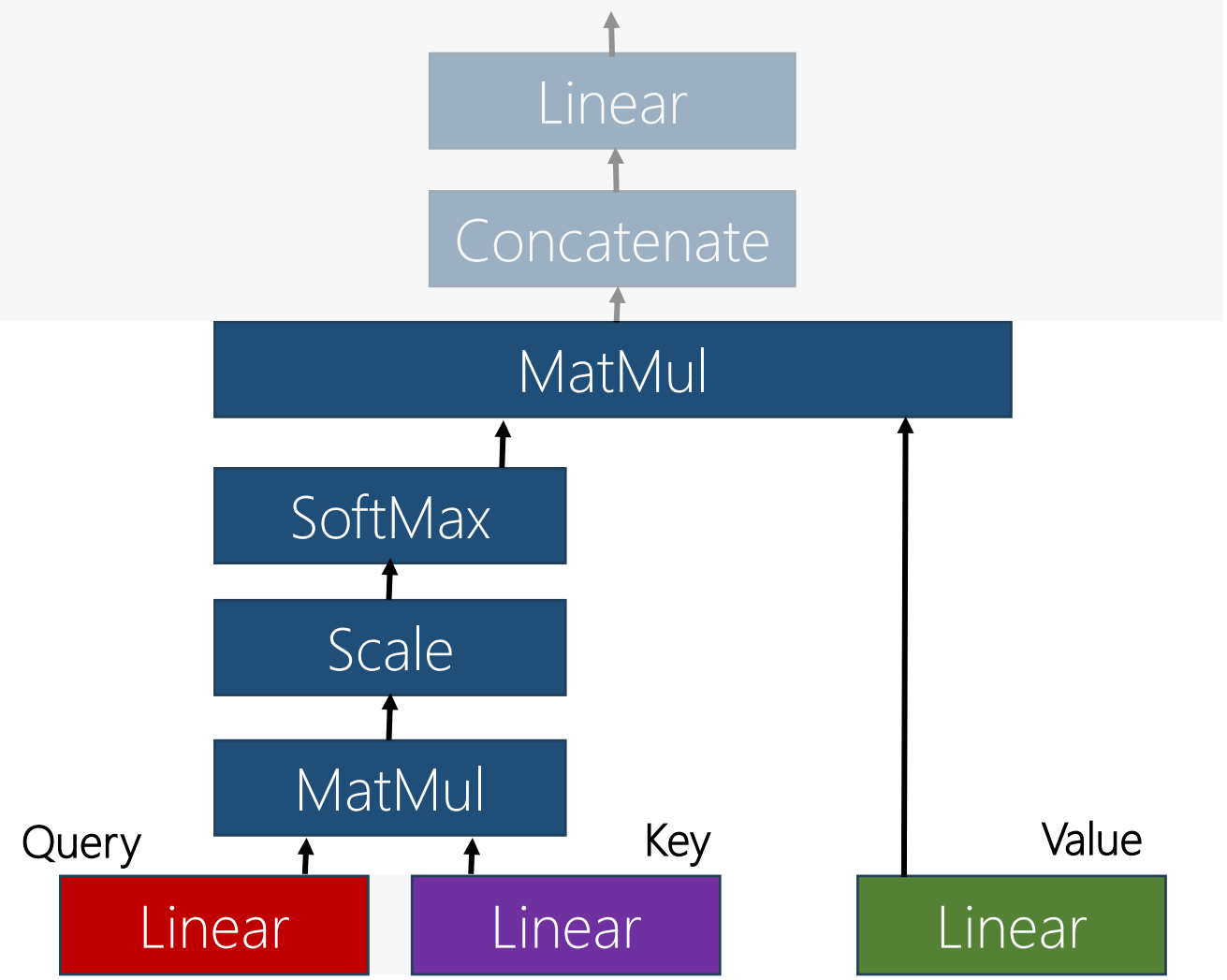


Normalize using SoftMax!

To bounded vals of 0 → 1

1 = high attention (high similarity)

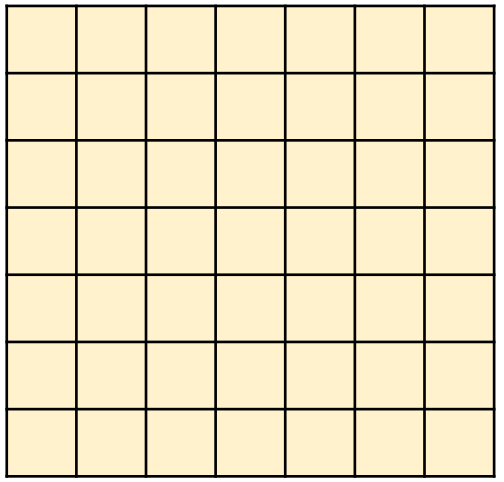
0 = not attended (not related at all)



Weighted Value

weights for how close each word is to each other

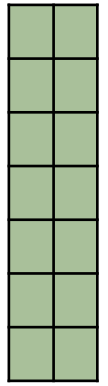
7 x 7



X

vector representing the words numerically

7 x 2



Normalized Attention Filter

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

query key

softmax val

numerator

val vector

7x2 matrix

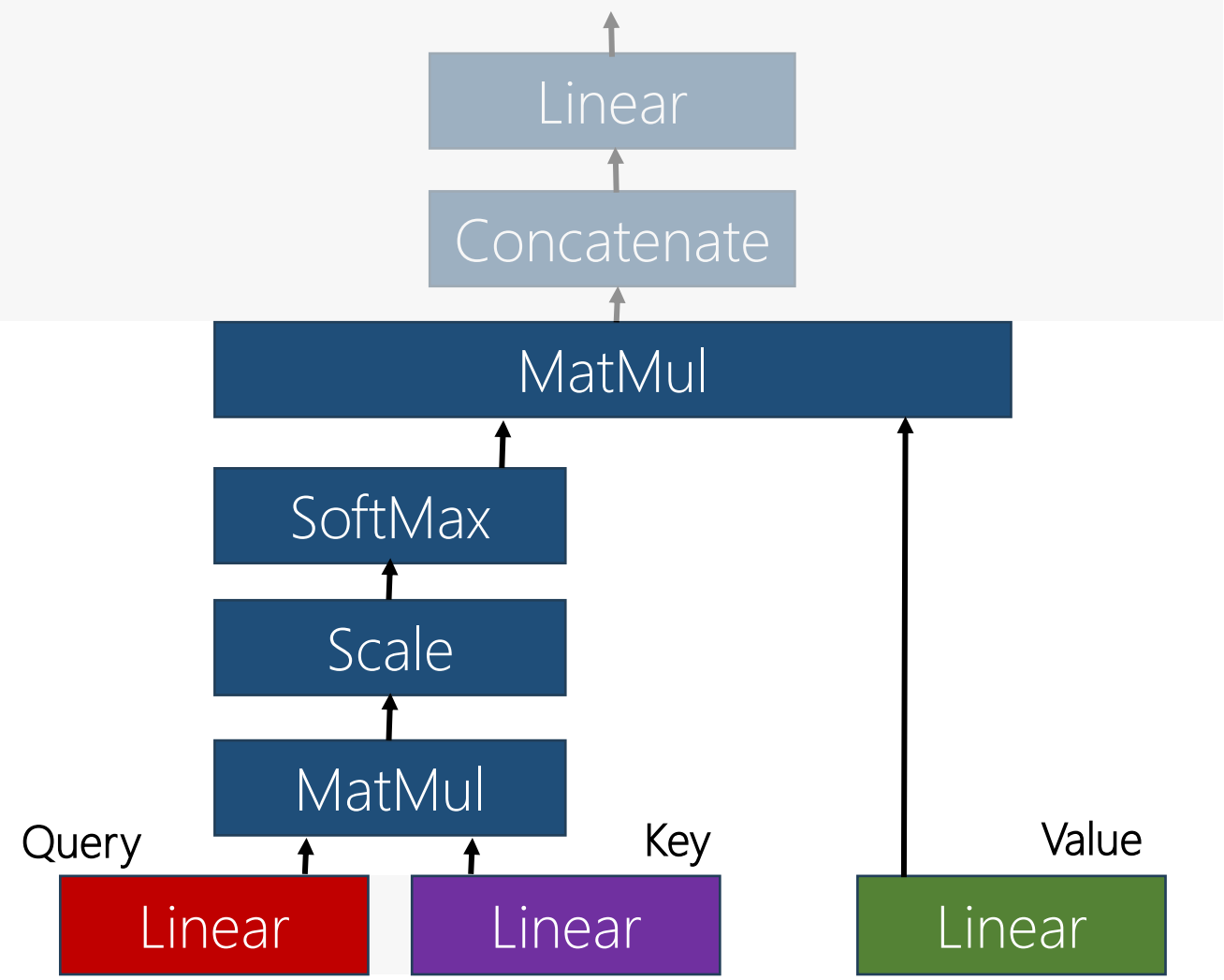
This is 1 attention heads

What is this output?

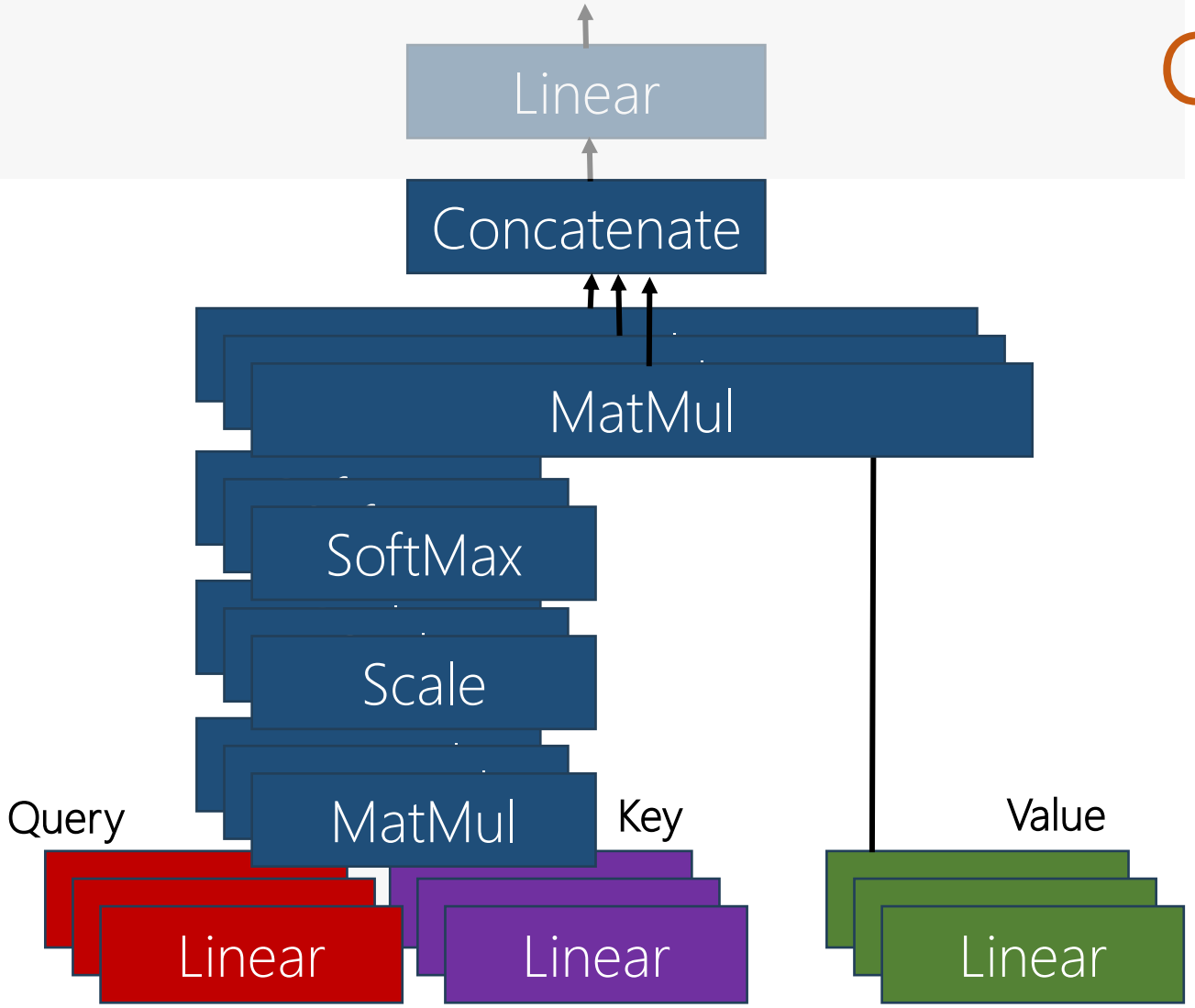
output is weighted words rep

What's to Concatenate??

We do the above multiple times to get multiple attention heads to give more learning capabilities

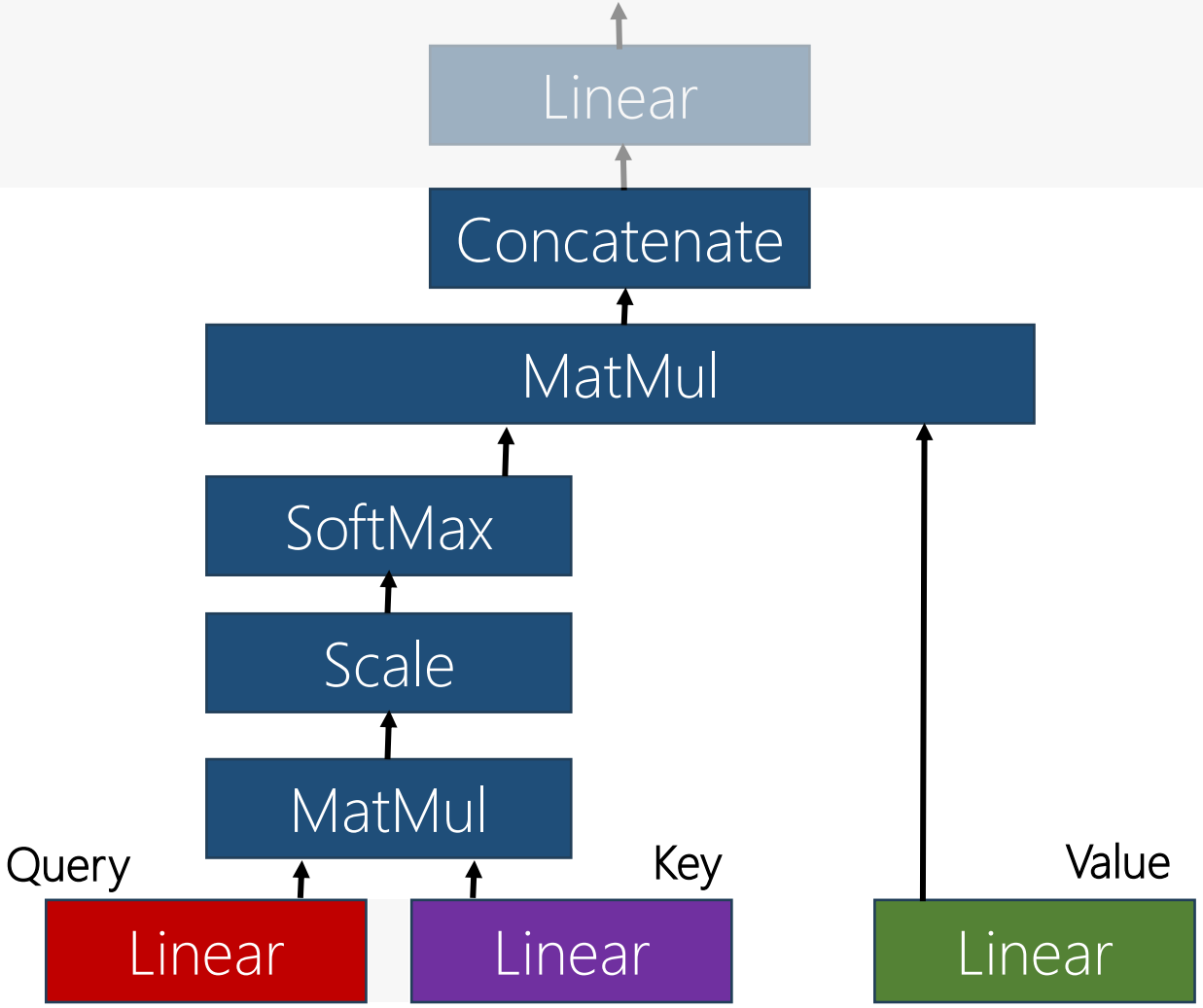


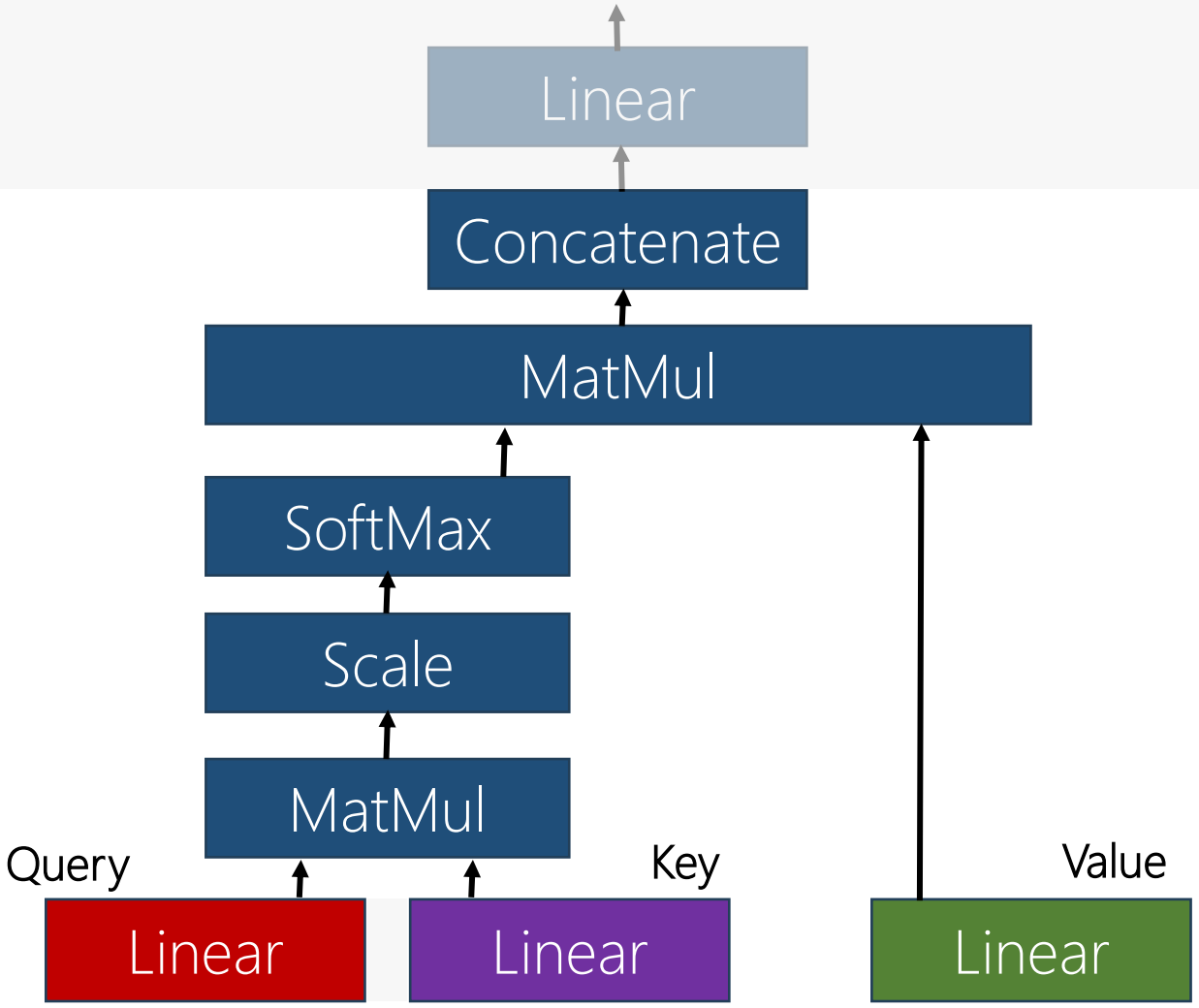
Concatenate Outputs from all Attention Heads



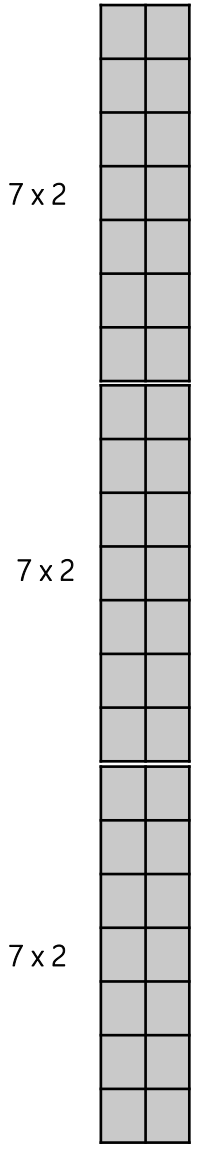
Each processed parallelly, concat all the attention outputs

Another Linear layer?? Why? Why? Why?

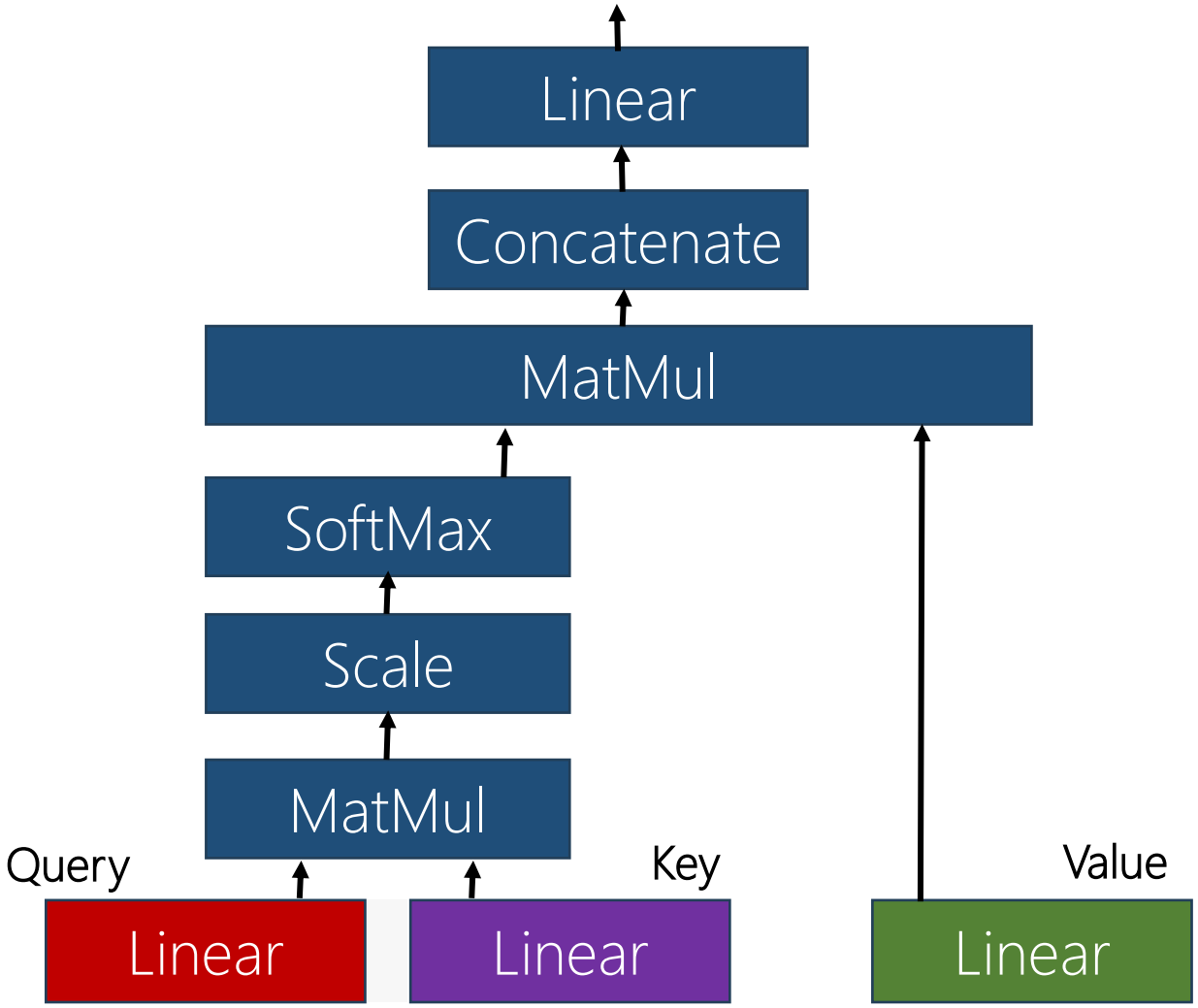




Another Linear layer?? Why? Why? Why?

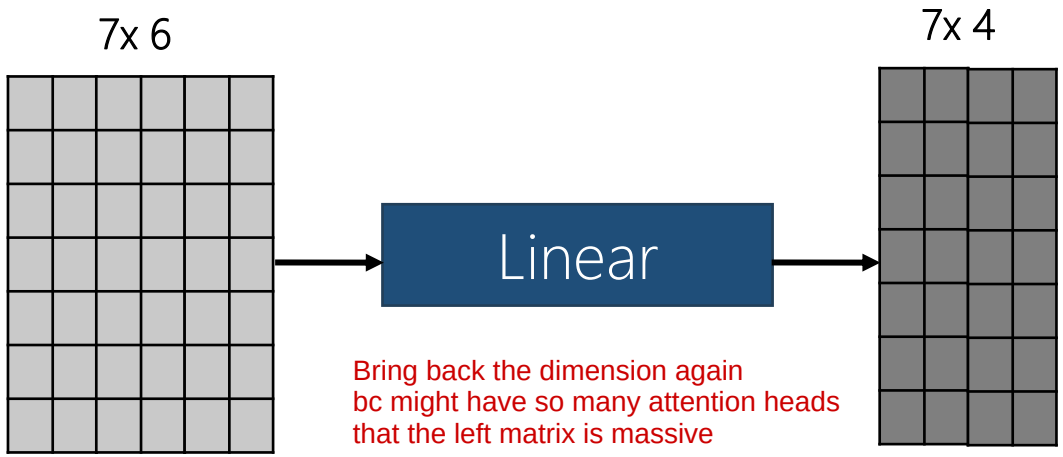


have many attention heads to combine



Another Linear layer?? Why? Why? Why?

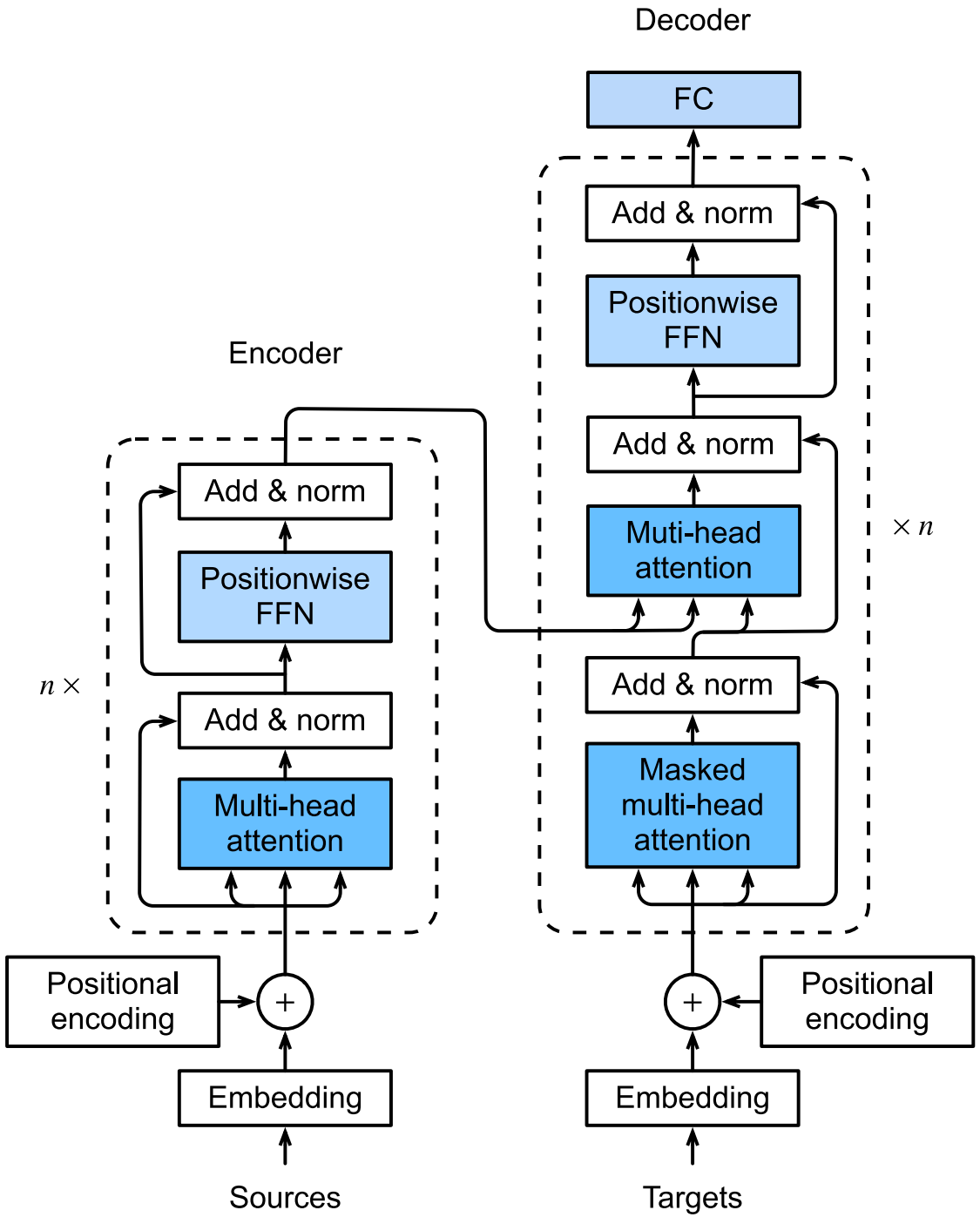
dimension back to something tracktable

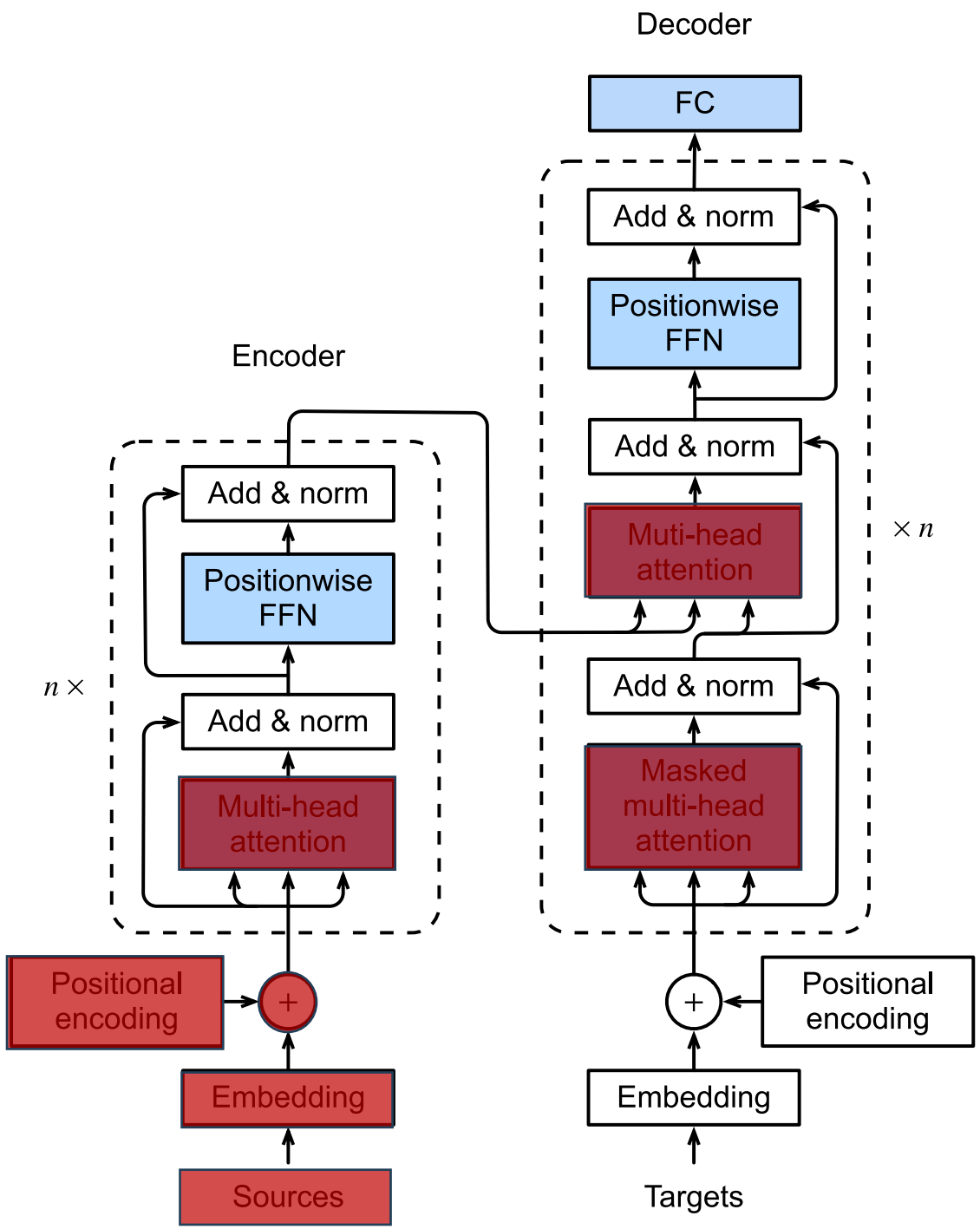


Bring back the dimension again
bc might have so many attention heads
that the left matrix is massive

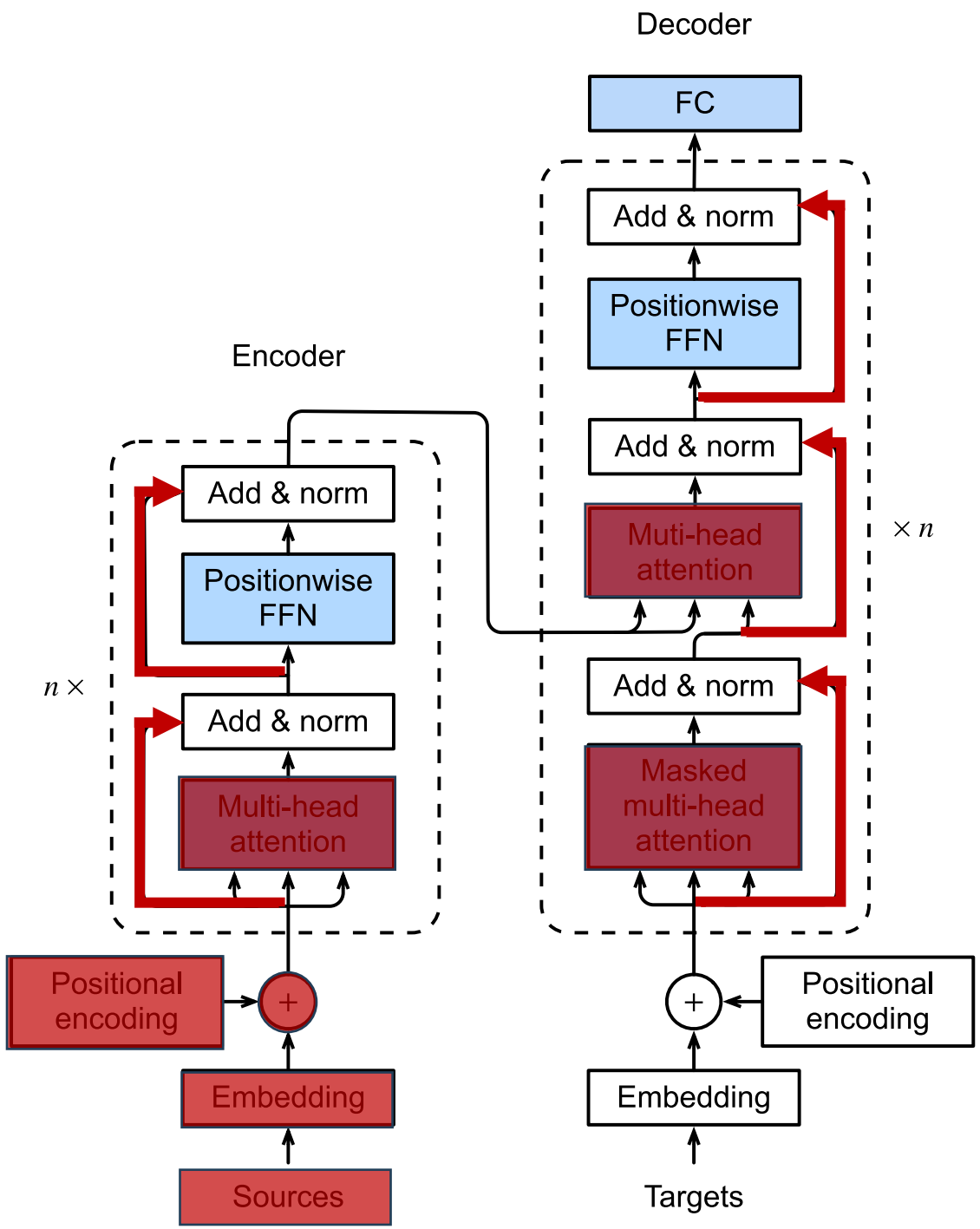
Continue here next time.....

What about the rest of the Transformer?


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Positional encoding
Embedding
Embedding
Sources
Targets


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
+
Embedding
Sources
+
Positional encoding
Embedding
Targets

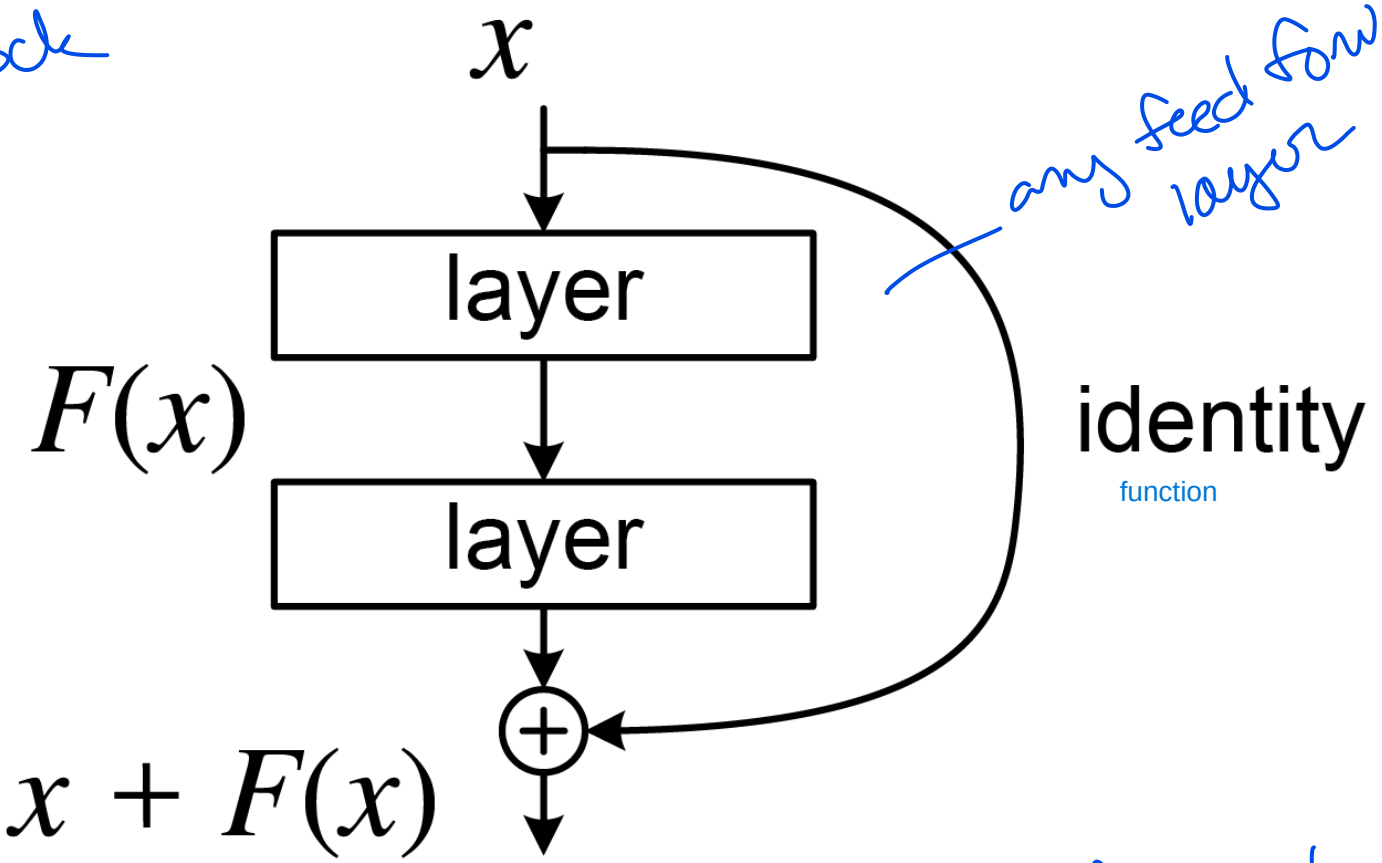
Residual Connection



Let's go down the memory lane...

Residual neural network

A single connection for 1 block



any feed forwarding layer

Remember backprop trying to learn weights: it's trying to learn F(x)-x as the residual

At every layer pass the original image and the transformed one

WHAT MATH WERE WE SUPPOSED TO DO??????

↳ once have transforms, add input info

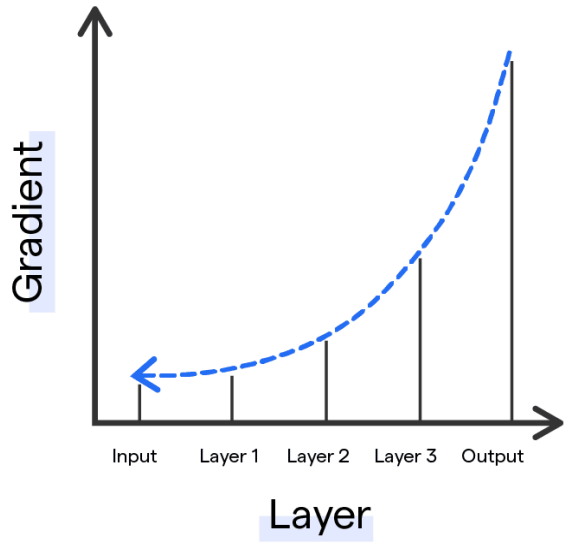
He et al. Deep residual learning for image recognition.CVPR, 2016

Vanishing Gradient Problem

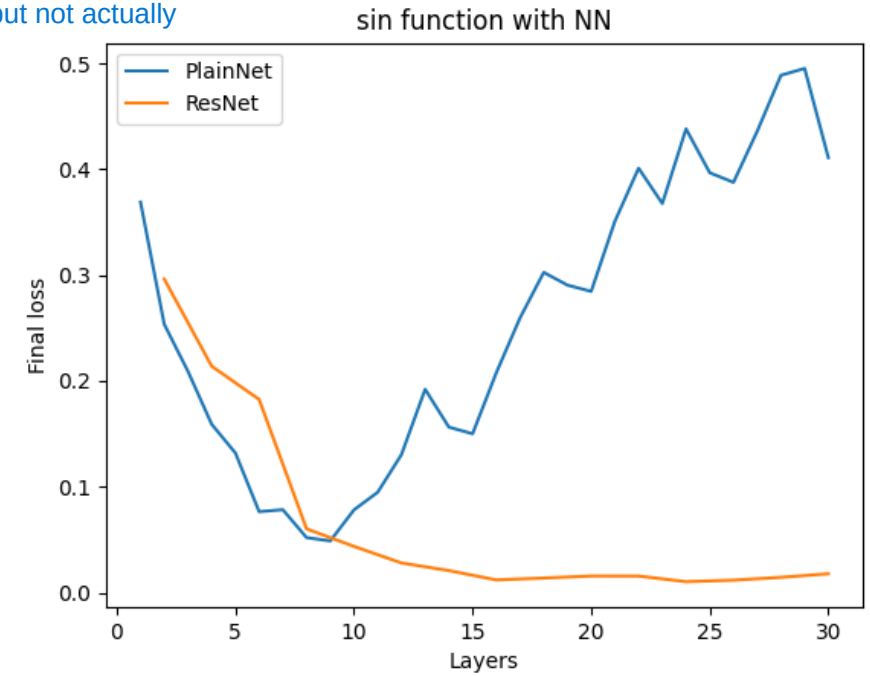
Residual neural network

If train a network with more and more layers it starts diverging instead of converging

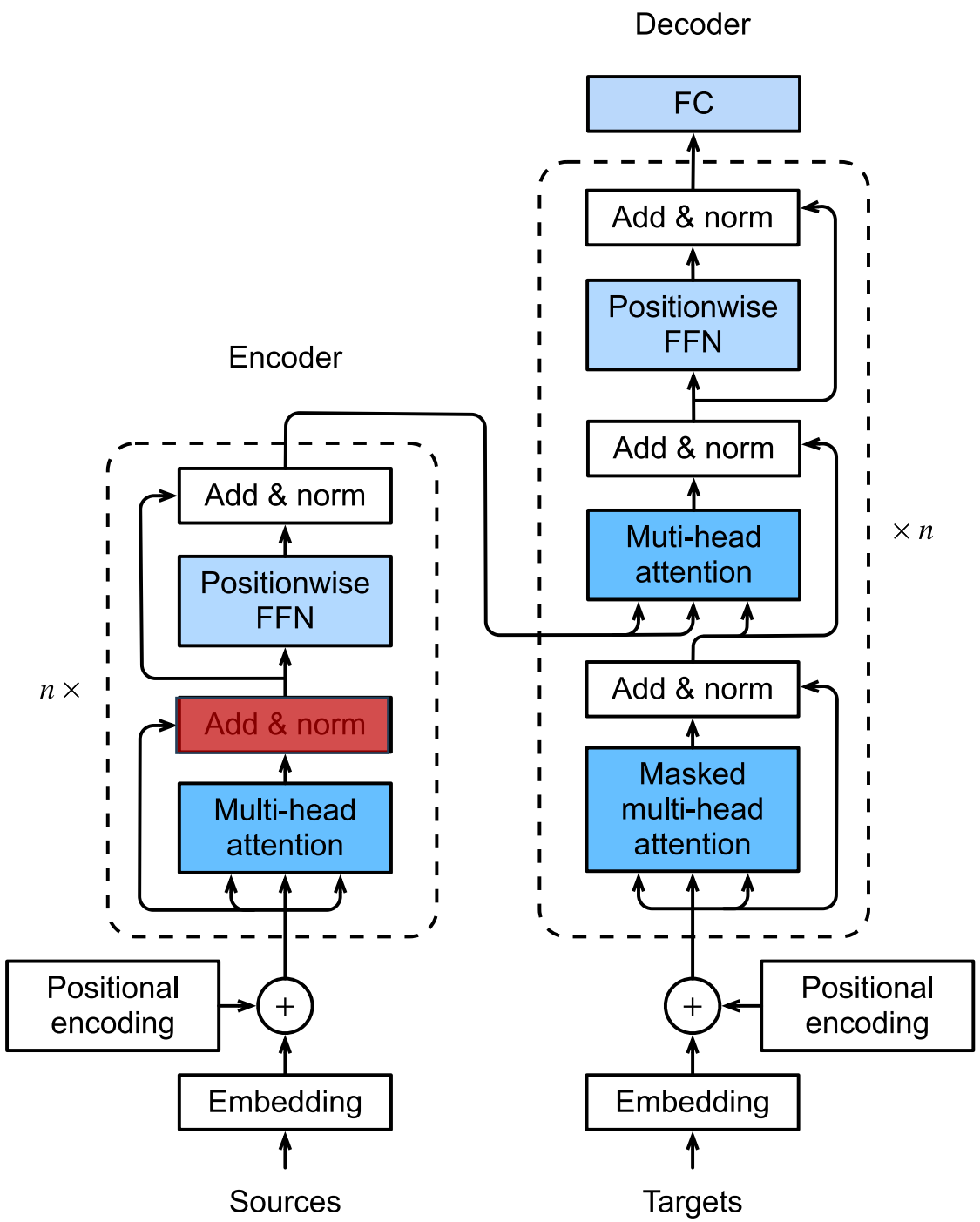
- Vanishing Gradient: As we backpropagate the gradients back through the network from the output layer towards the input the repeated multiplication of these small derivative values leads to increasingly smaller gradients could be close to zero but not actually



- Degradation problem: As a network deepens, the accuracy can start to decline even when training properly, which is not simply overfitting

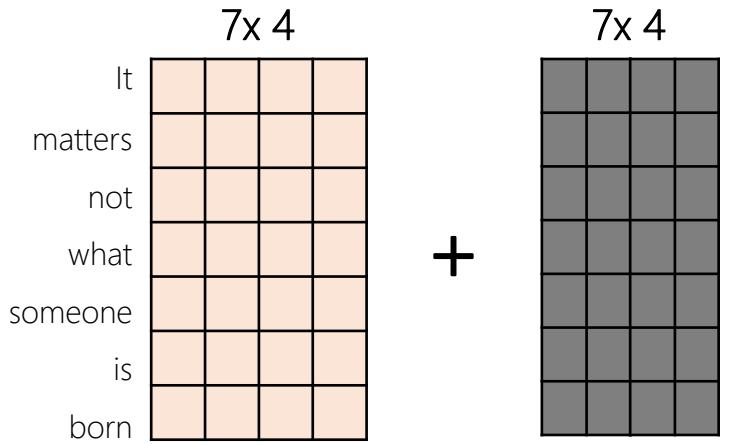


He et al. Deep residual learning for image recognition.CVPR, 2016



Add & Norm

What are we adding?

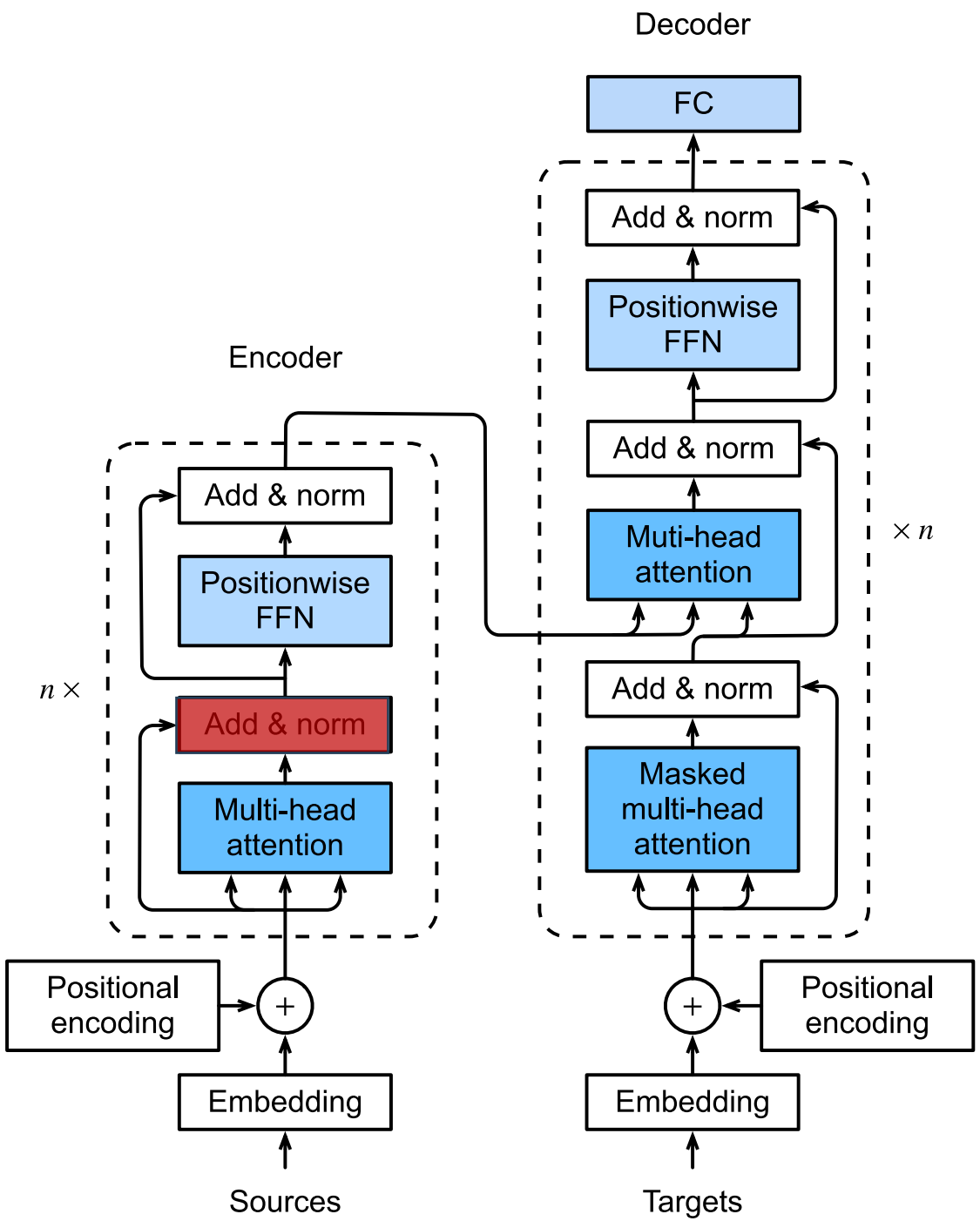


Word Emb. + Position Emb.

Multi-head Output

Ppl now do layer normalization not batch normalization to fix covariance shifts

Pytorch has a layer norm function

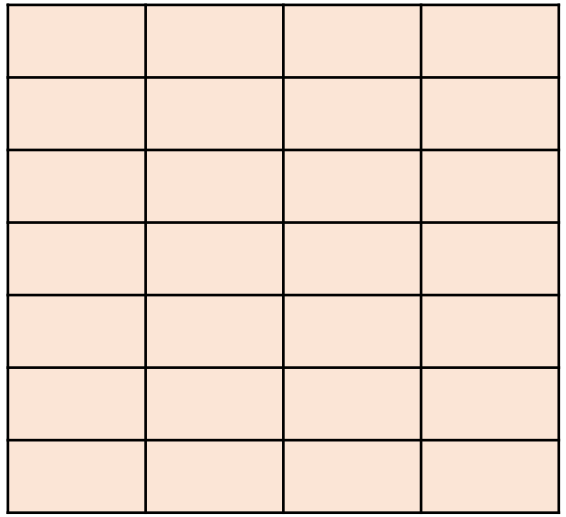


Add & Norm

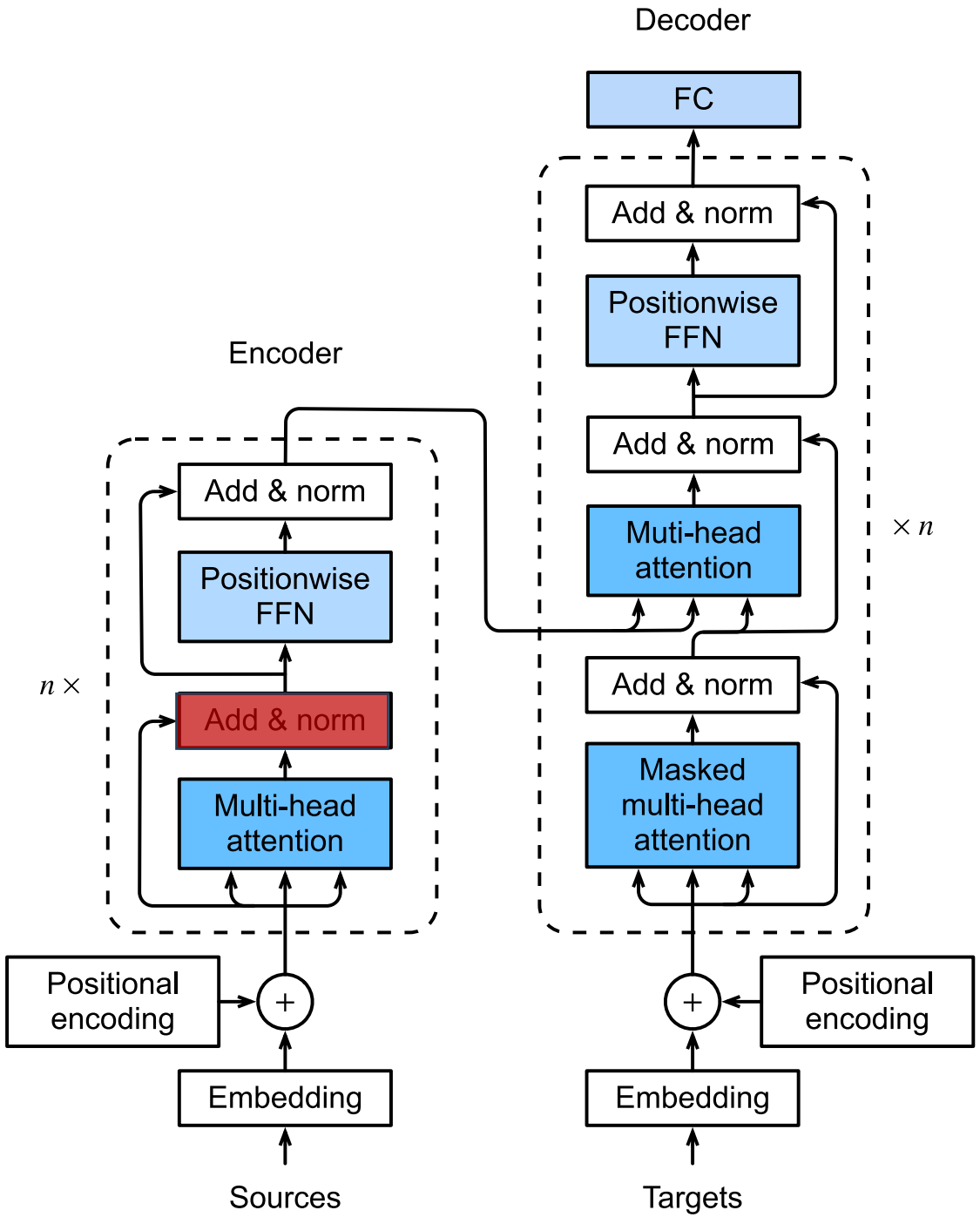
Solves internal covariance shift inside the model representation

LayerNorm is used to stabilize the training process and addresses the internal covariate shift (ICS) problem, where the distribution of activations within a layer changes during training, making it difficult for the network to learn effectively.

7x 4



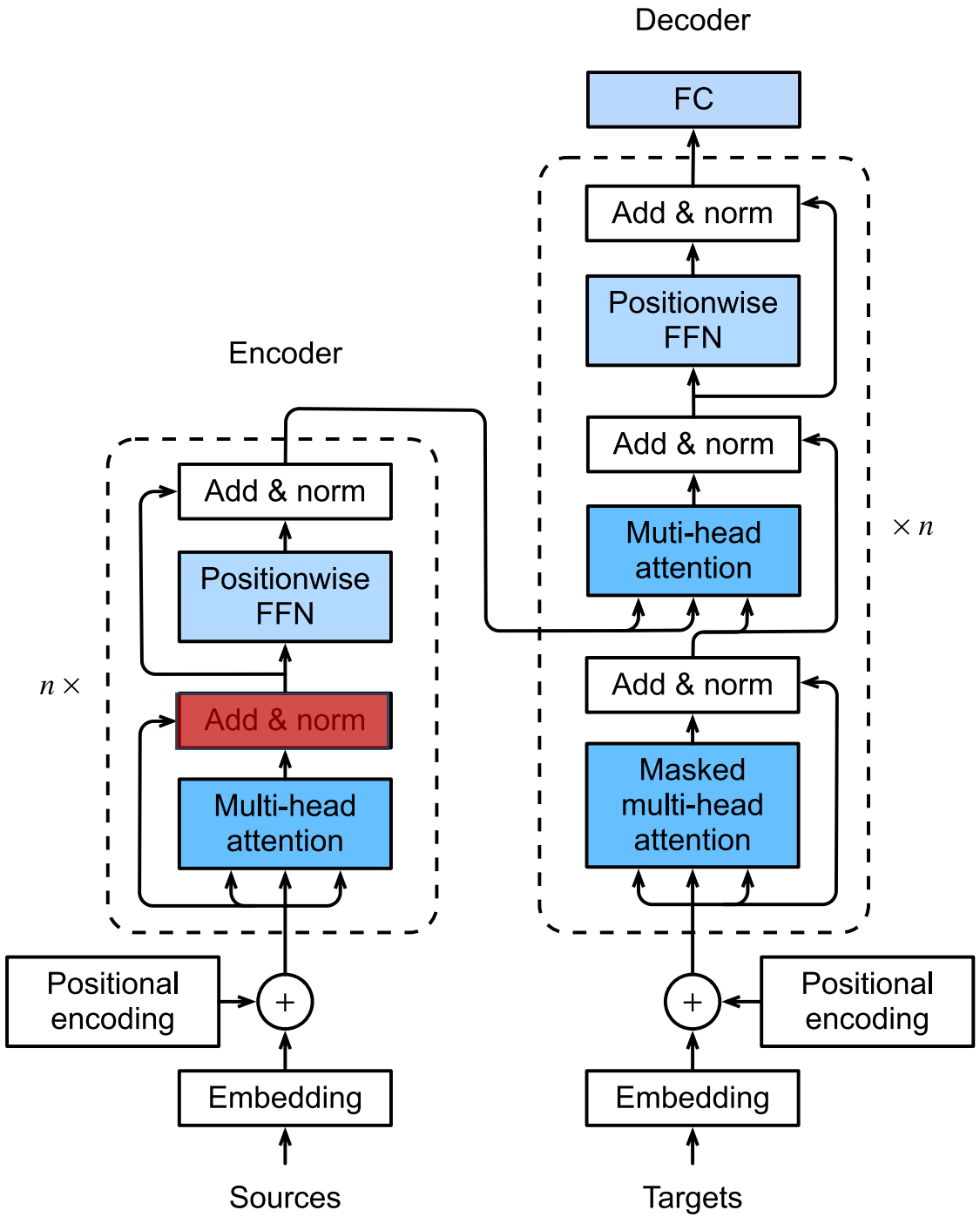
Add & Norm



Attention output:

7x 4

It	1.56	2.12	0.91	2.87
matters	0.45	1.23	2.76	0.67
not	2.03	0.58	1.41	1.29
what	0.92	2.31	0.14	2.55
someone	1.80	0.61	2.98	1.52
is	2.67	0.33	1.99	1.74
born	0.48	2.40	1.68	0.29



Add & Norm

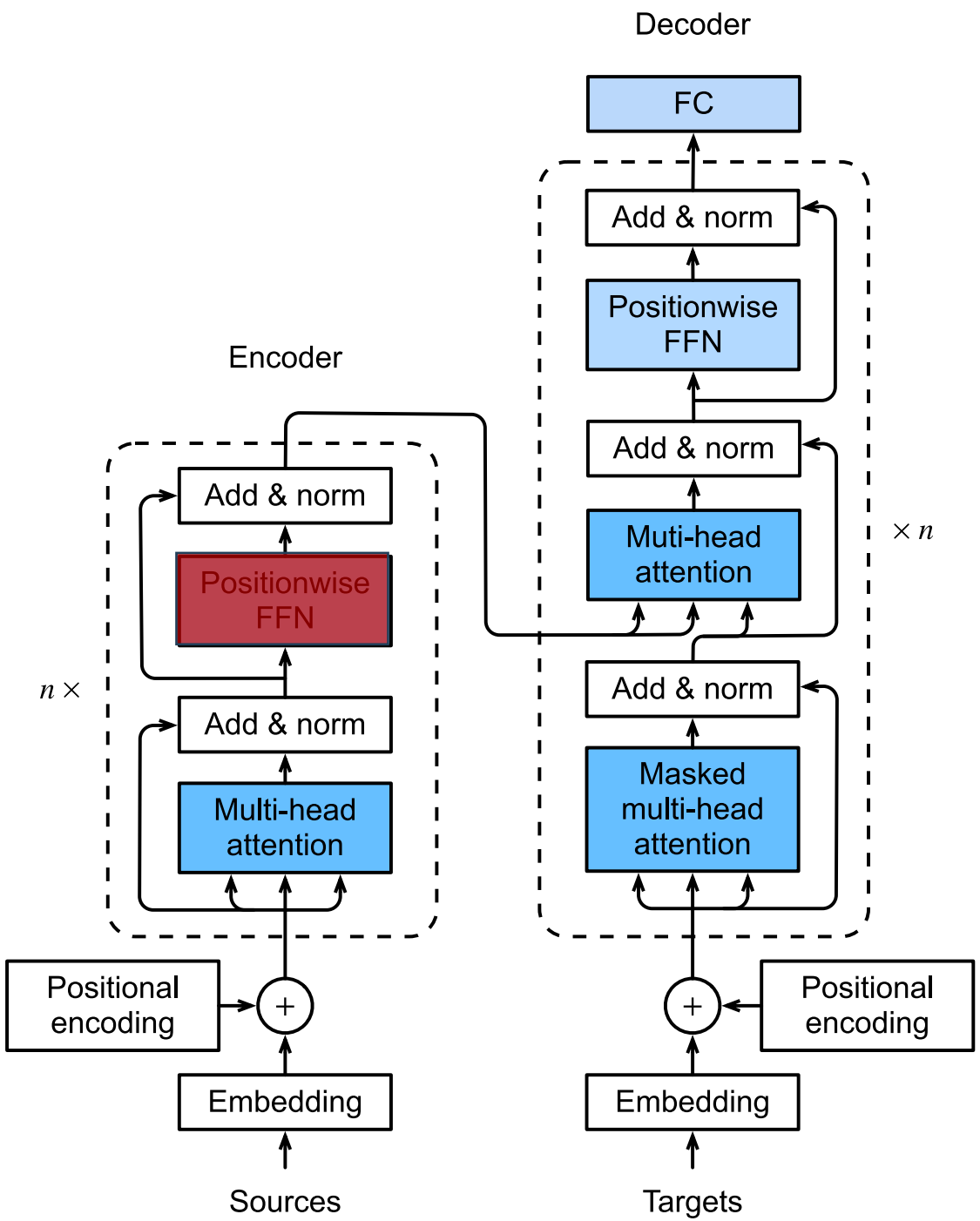
- add the layers and normalize
- calc mean and stdv row wise, want normalization of representation of each word
- 7 is num words, 4 is embedding dimension

7 x 4

					Mean (μ)	Stdev. (σ)
It	1.56	2.12	0.91	2.87	1.865	0.673
matters	0.45	1.23	2.76	0.67	1.028	0.867
not	2.03	0.58	1.41	1.29	1.578	0.504
what	0.92	2.31	0.14	2.55	1.730	0.838
someone	1.80	0.61	2.98	1.52	1.978	0.811
is	2.67	0.33	1.99	1.74	1.930	0.656
born	0.48	2.40	1.68	0.29	1.198	0.822

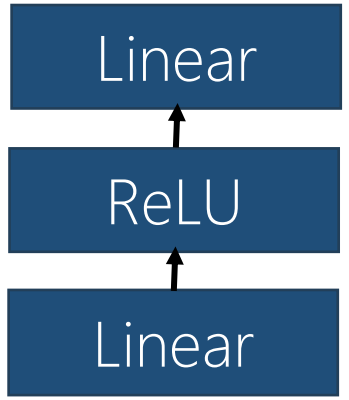
normalize each value in the attention matrix ?

$$x_i' = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \varepsilon}}$$

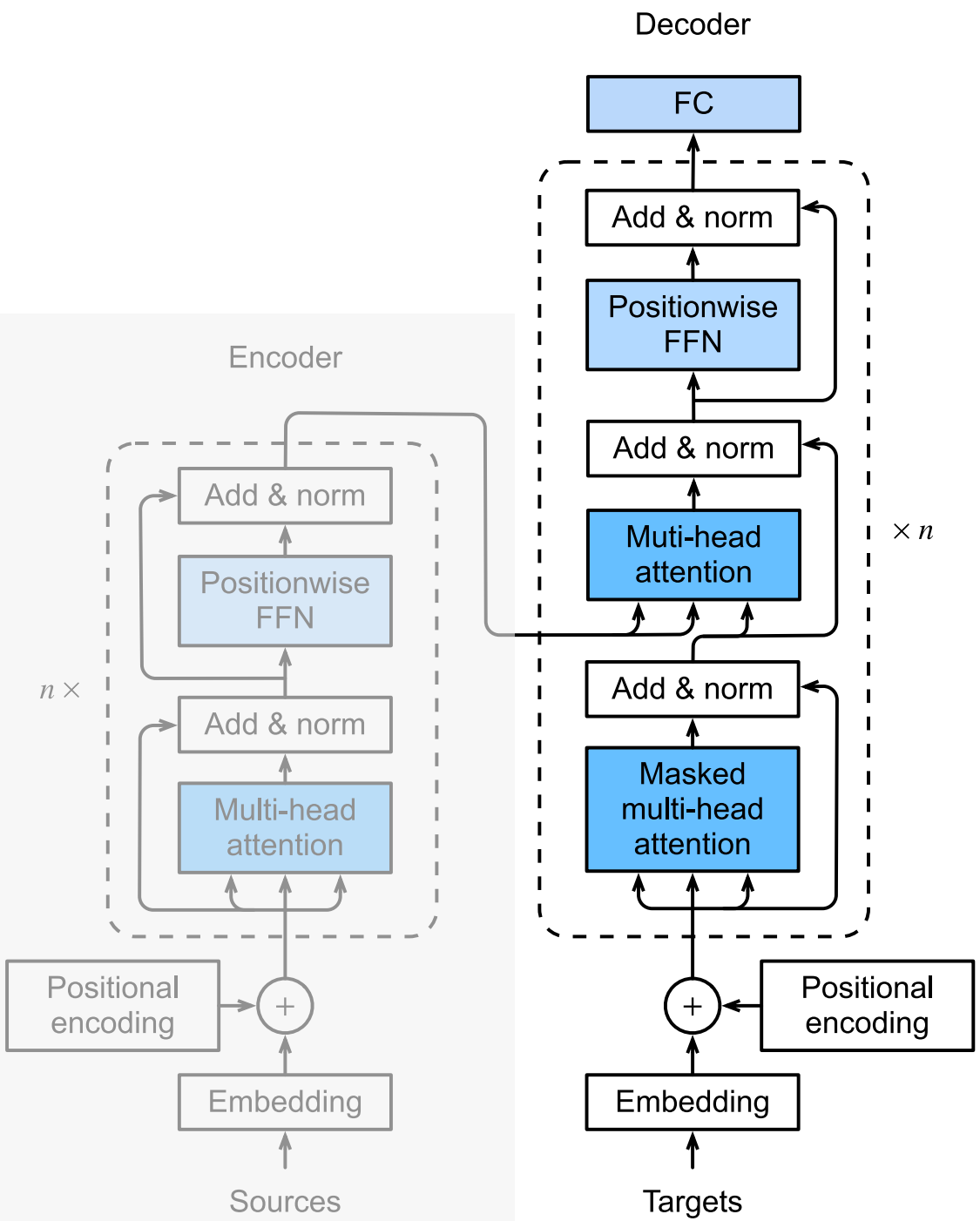


A bunch of Linear is all we need!

Now use GeLU

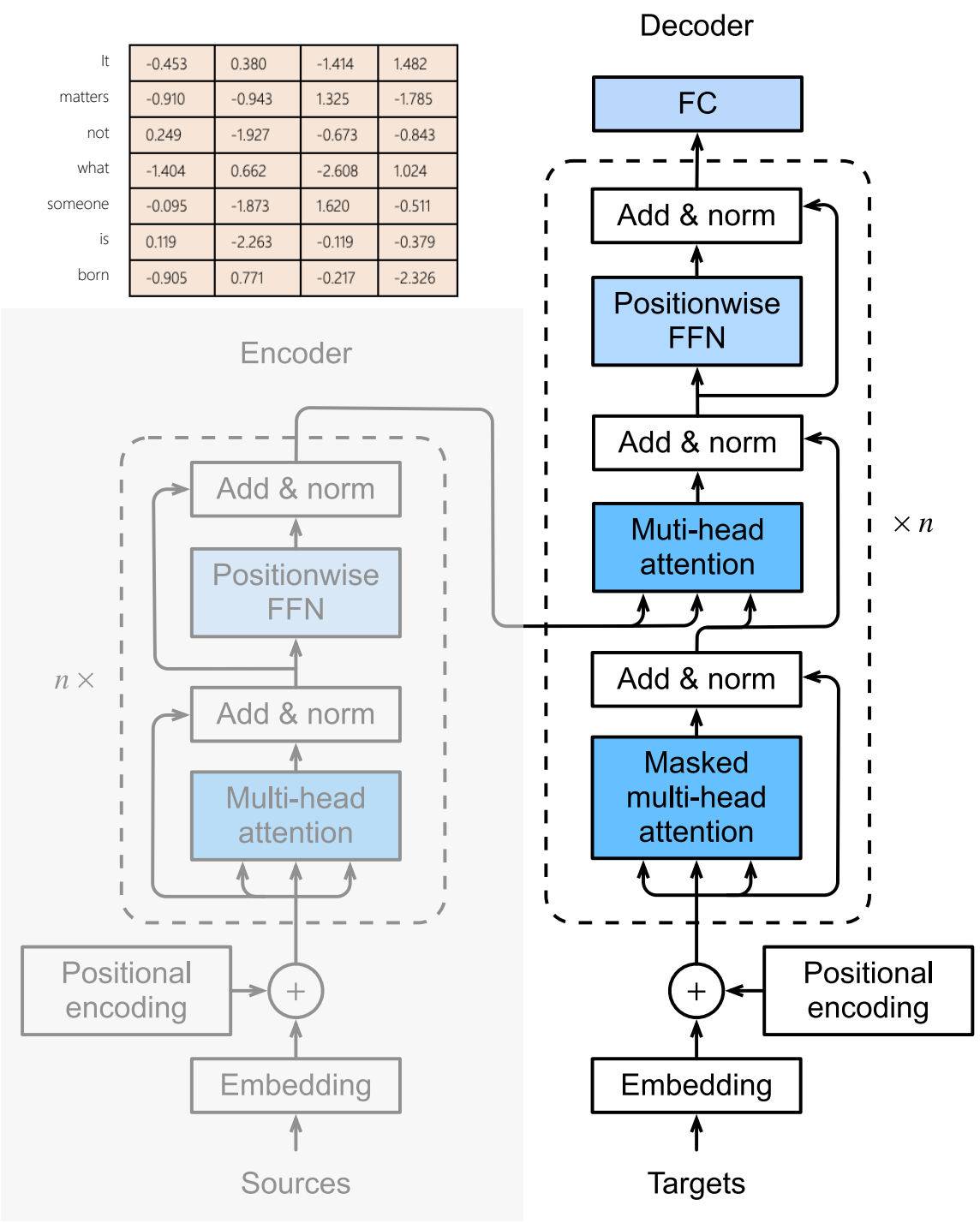


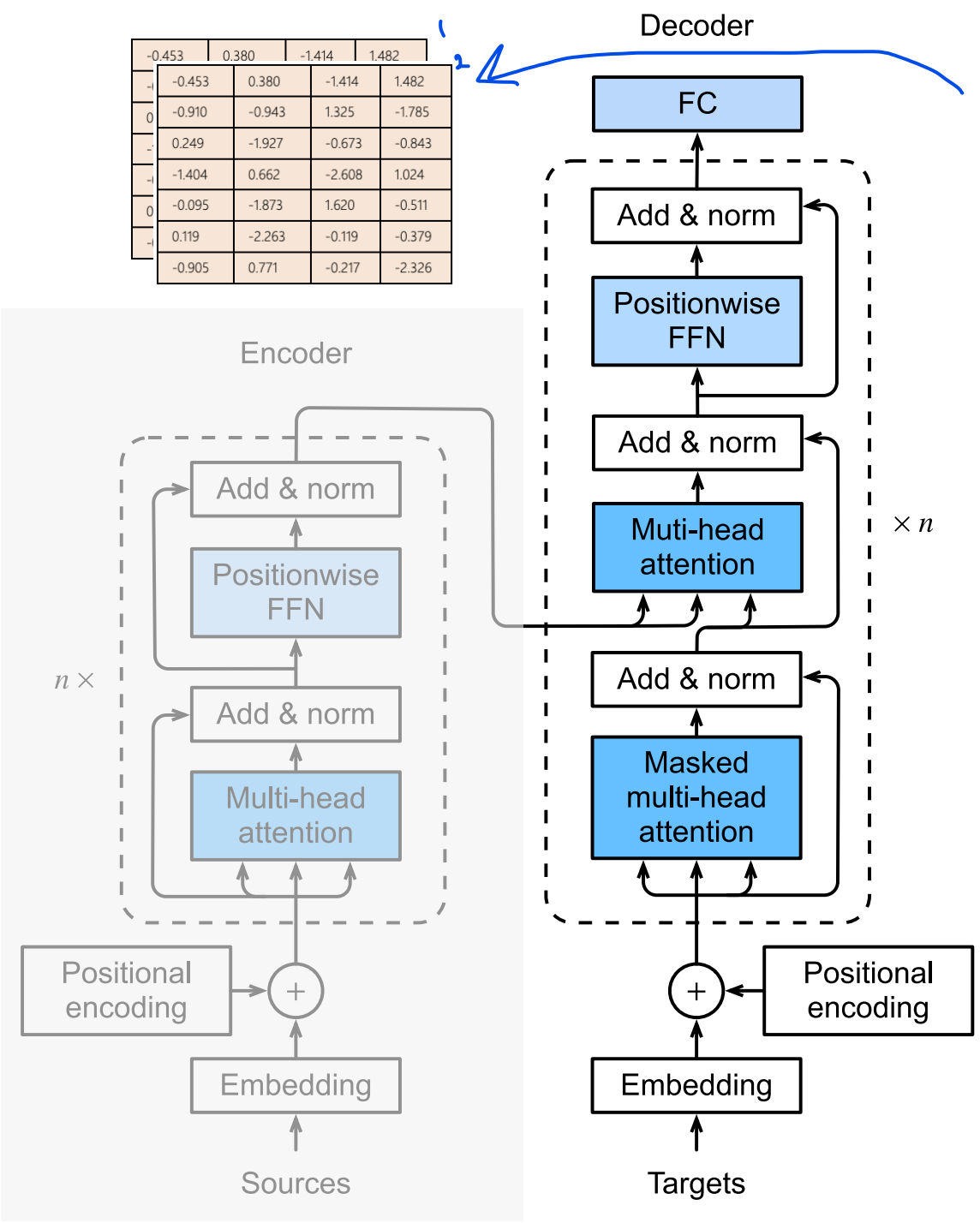
→ activation function (could use another)



That covers our Encoder part of the Transformer!

It	-0.453	0.380	-1.414	1.482
matters	-0.910	-0.943	1.325	-1.785
not	0.249	-1.927	-0.673	-0.843
what	-1.404	0.662	-2.608	1.024
someone	-0.095	-1.873	1.620	-0.511
is	0.119	-2.263	-0.119	-0.379
born	-0.905	0.771	-0.217	-2.326





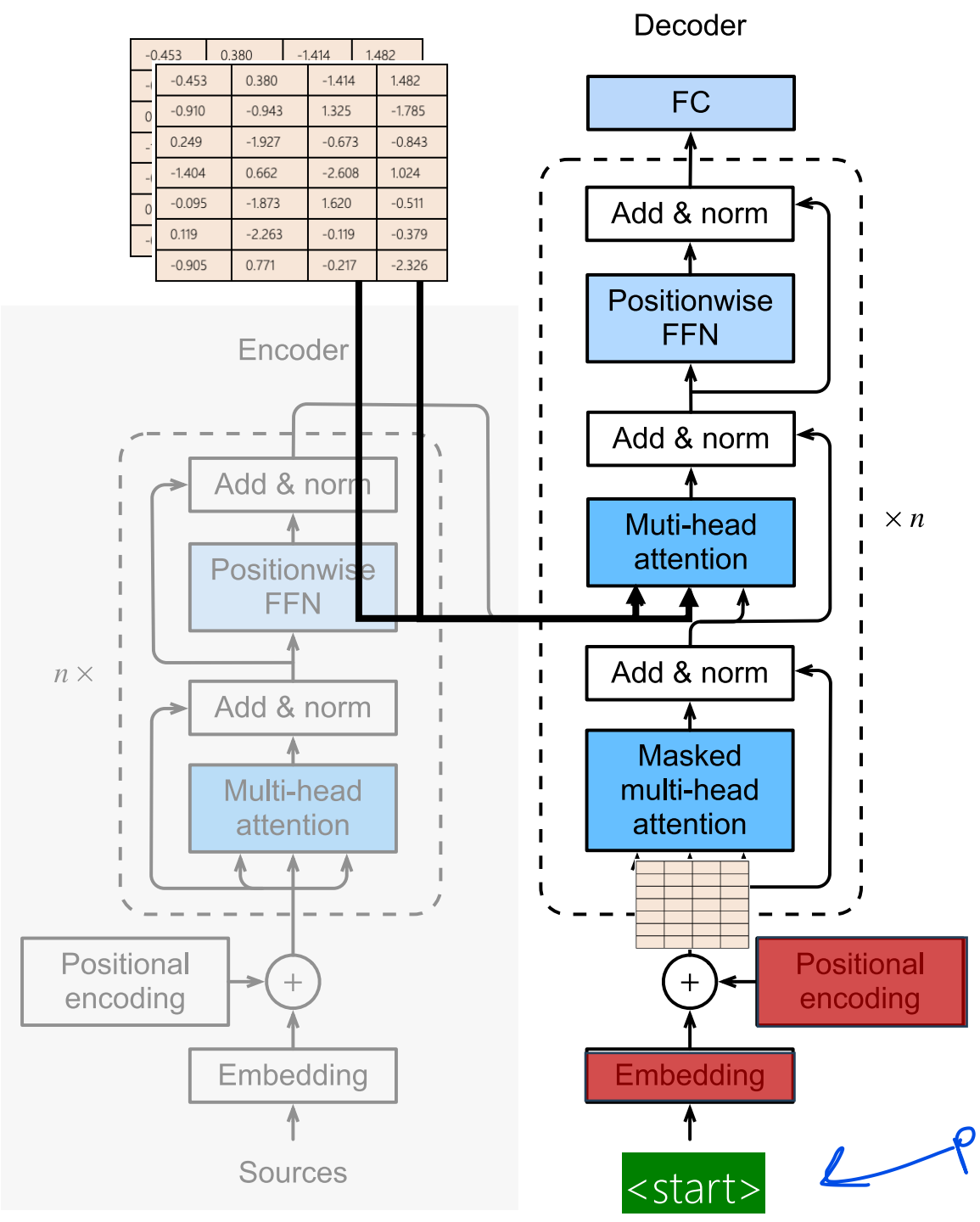
Create 2 copies of it

<start> - starts w/ "start" token, ends with "end" token

So it learns when to start/stop

Timestamp = 1

- Pt is to ask the model to complete the phrase

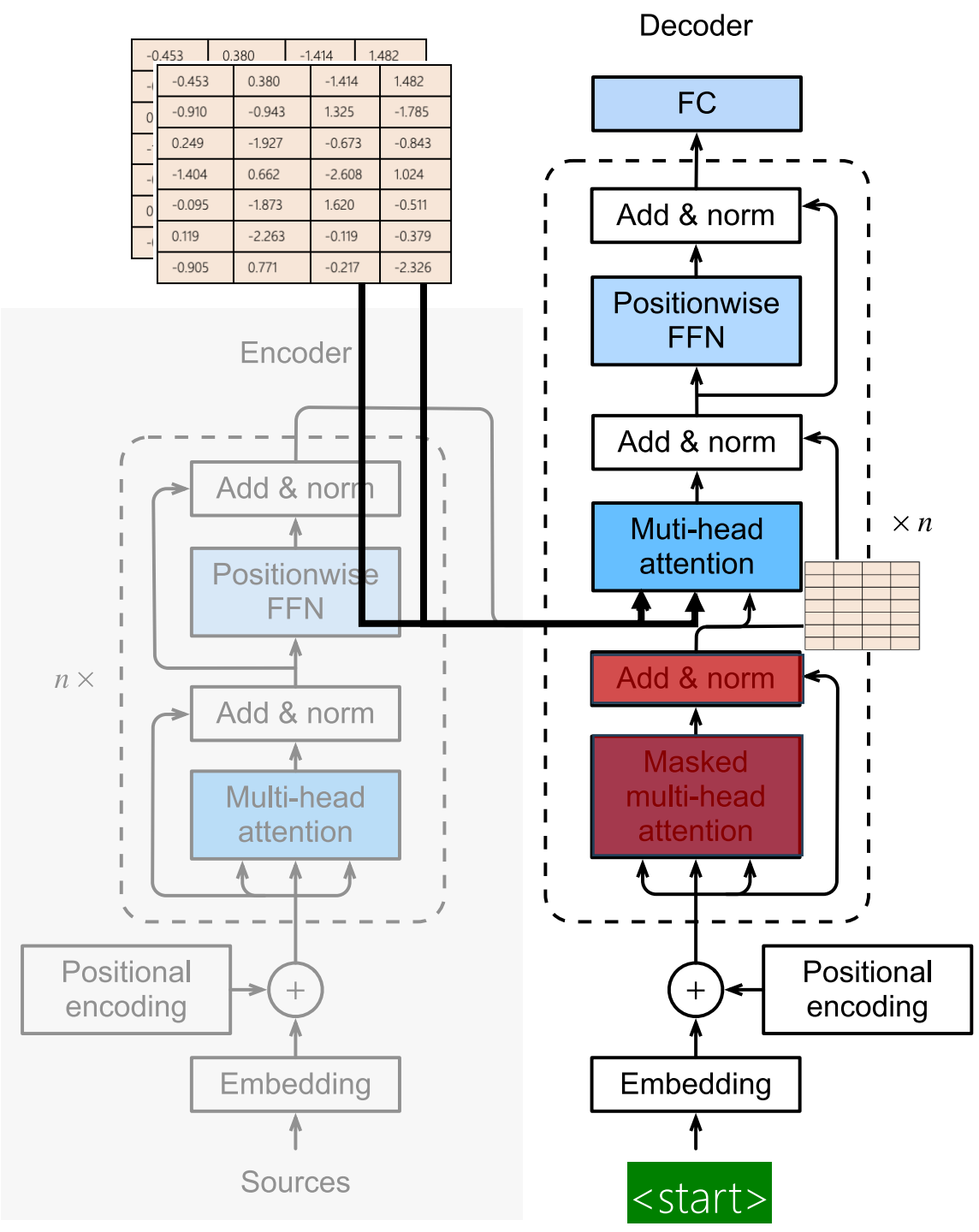


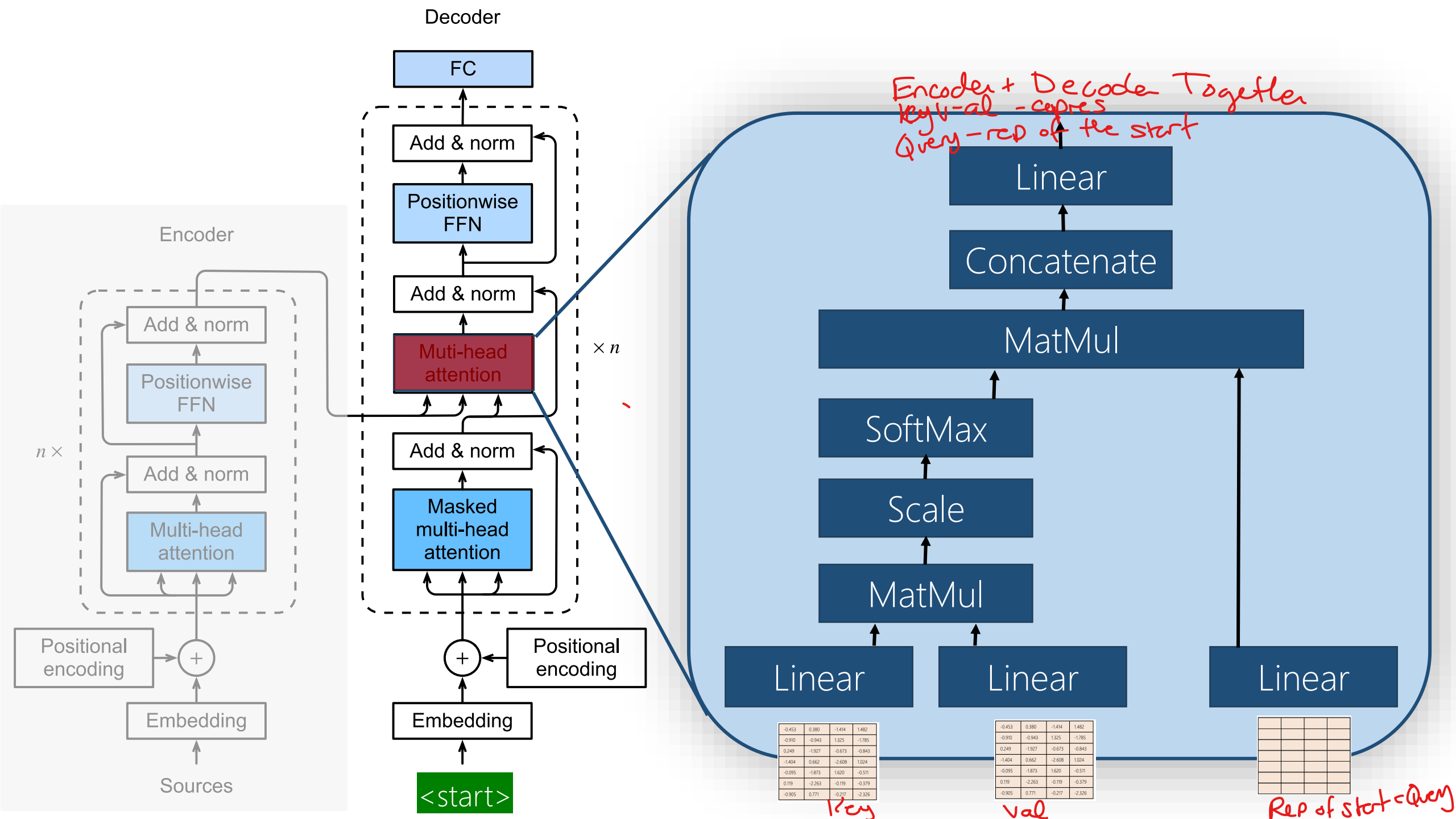
pass to ask it to start thinking of an answer

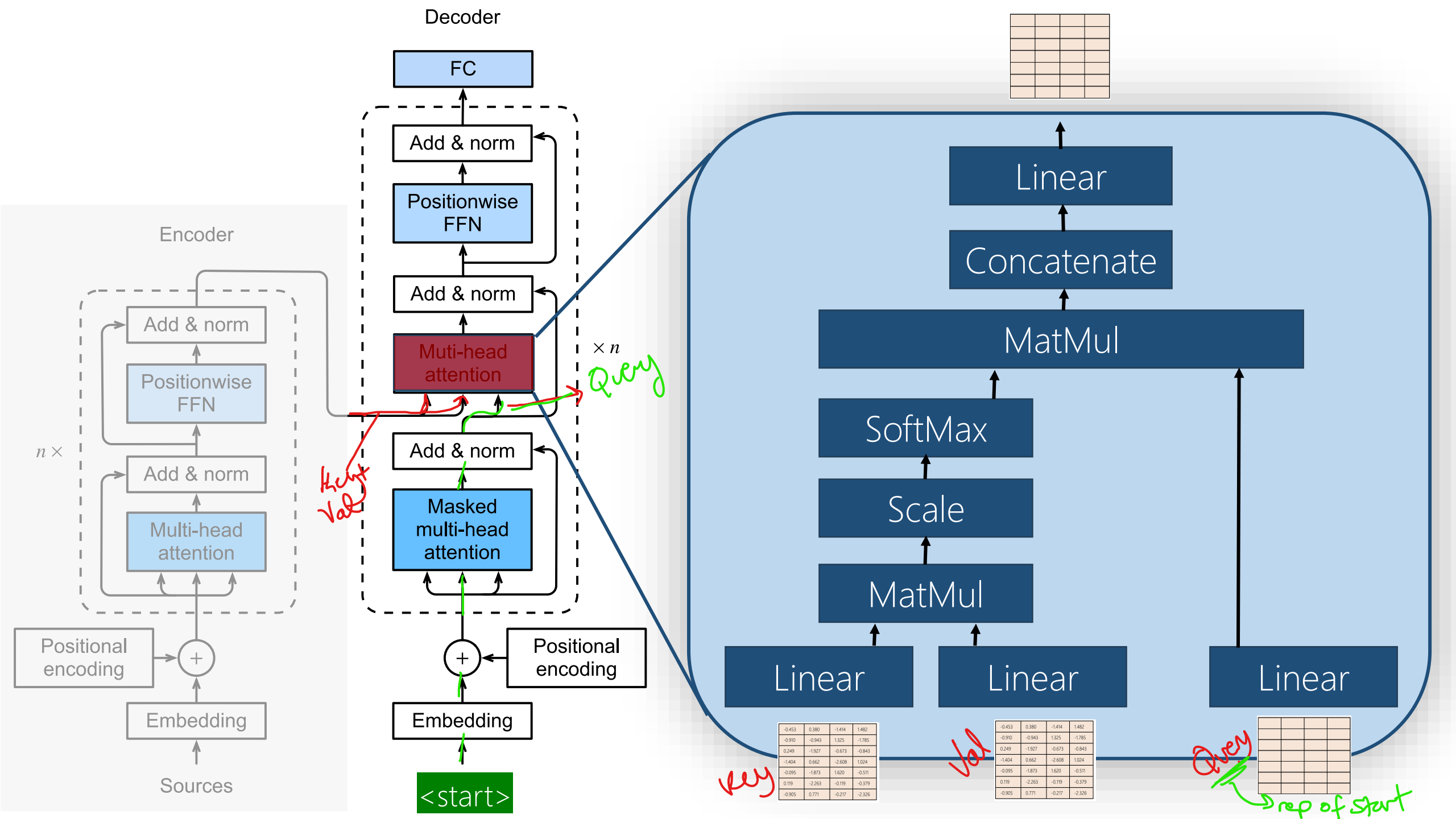
Timestamp = 1

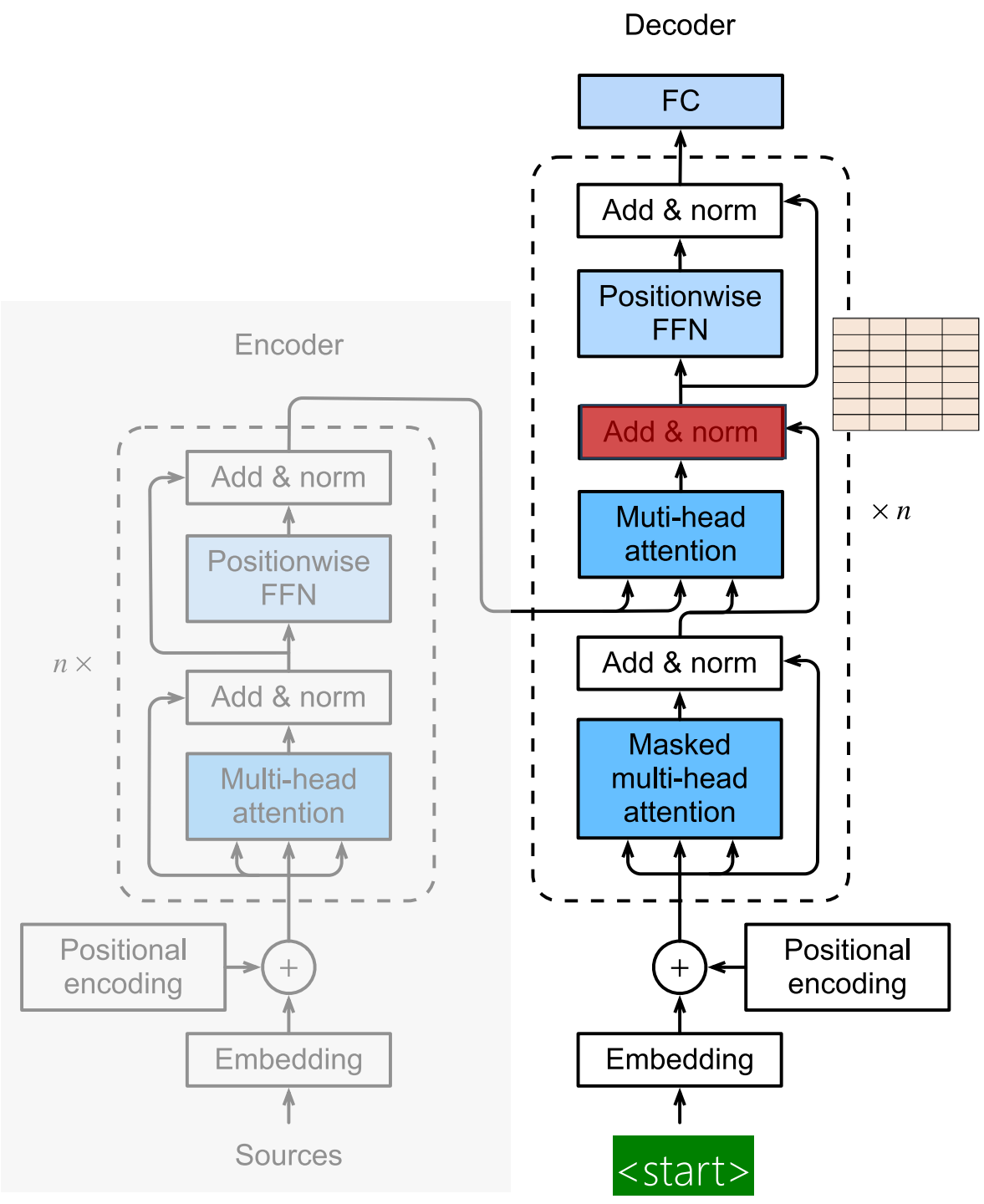
-0.453	0.380	-1.414	1.482
-0.910	-0.943	1.325	-1.785
0.249	-1.927	-0.673	-0.843
-1.404	0.662	-2.608	1.024
-0.095	-1.873	1.620	-0.511
0.119	-2.263	-0.119	-0.379
-0.905	0.771	-0.217	-2.326

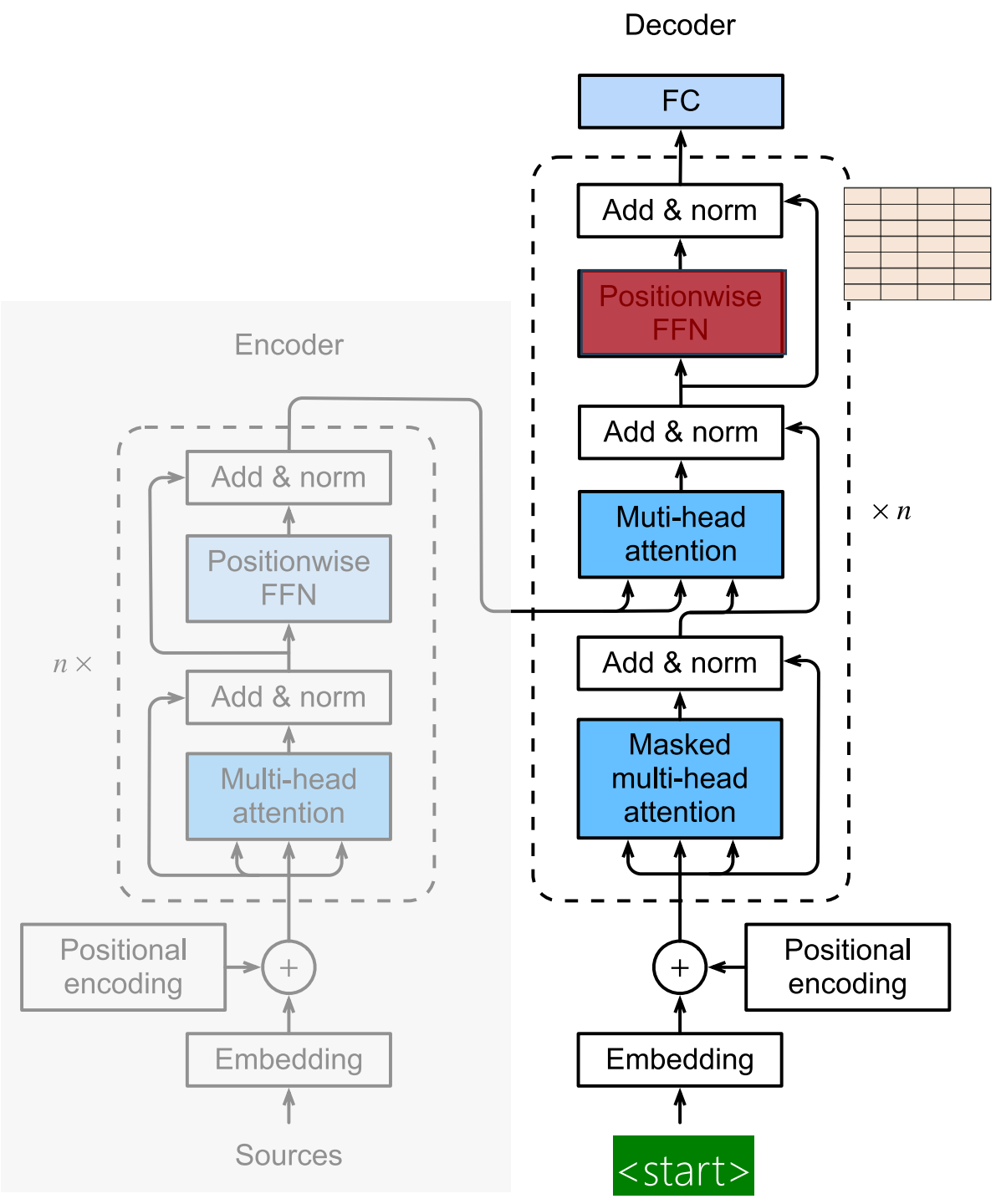
Timestamp = 1

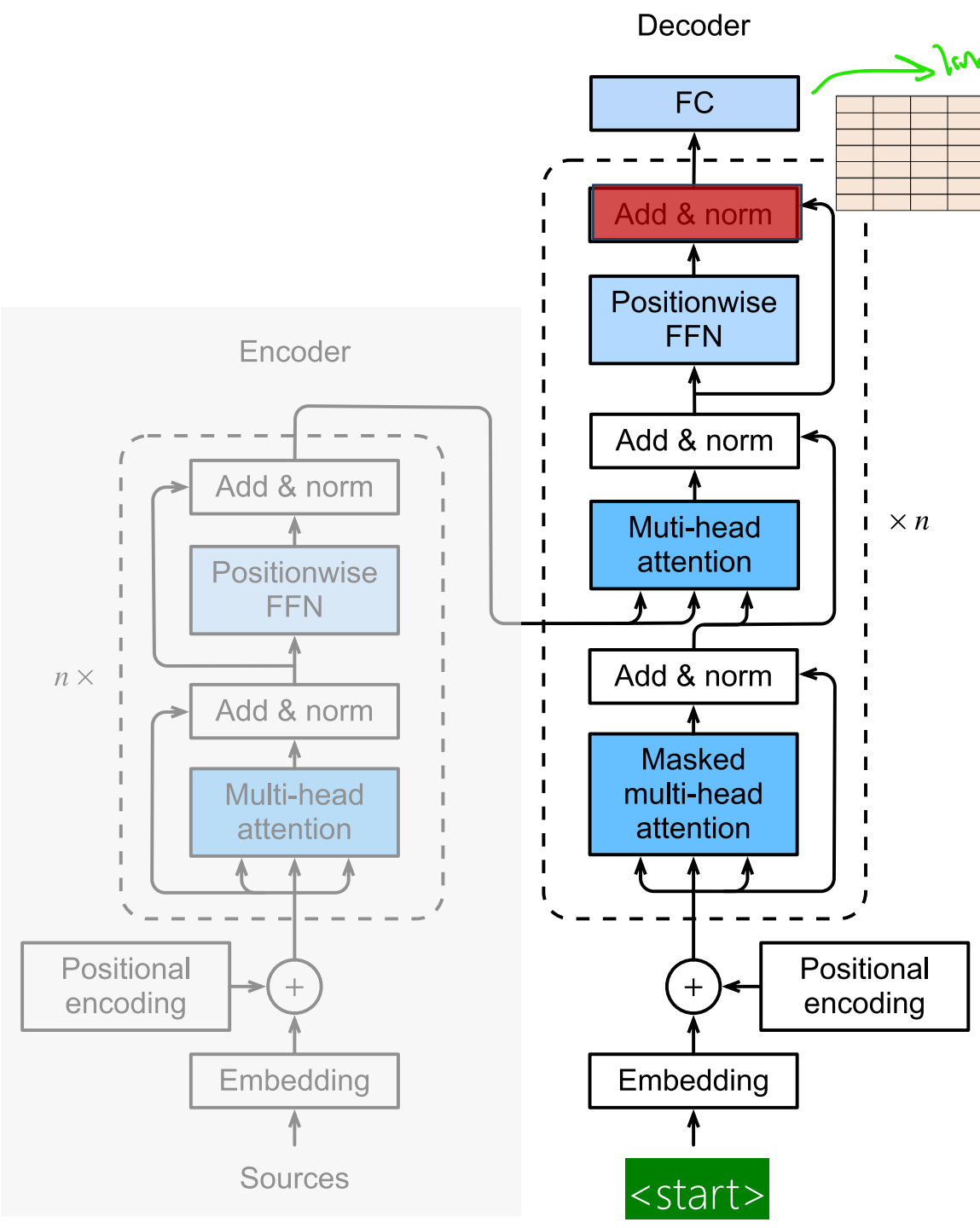

Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
× n
Add & norm
Masked multi-head attention
Positional encoding
Embedding
<start>
Encoder
Add & norm
Positionwise FFN
n ×
Add & norm
Multi-head attention
Positional encoding
Embedding
Sources


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Positional encoding
Embedding
<start>
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Embedding
Sources
Encoder + Decoder Together
key-val - copies
Query - rep of the start
Linear
Concatenate
MatMul
SoftMax
Scale
MatMul
Linear
Linear
Linear
Key
Val
Rep of start = Query


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
× n
Query
Add & norm
Masked multi-head attention
Positional encoding
Embedding
<start>
Encoder
n ×
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
Positional encoding
Embedding
Sources
Linear
Concatenate
MatMul
SoftMax
Scale
MatMul
Linear
Linear
Linear
key
Val
Query
rep of start


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Positional encoding
Embedding
<start>
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Embedding
Sources

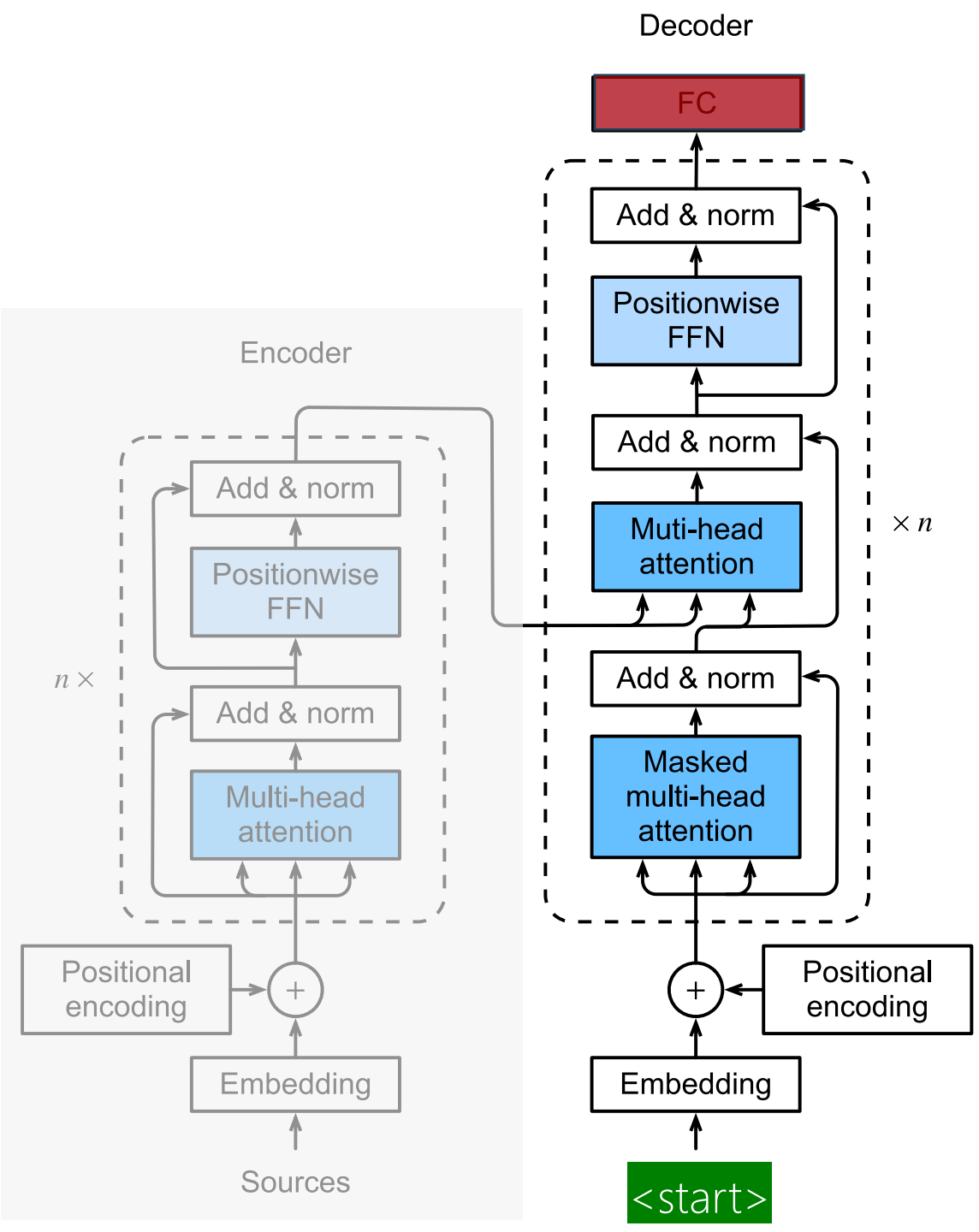

Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Positional encoding
Embedding
<start>
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Embedding
Sources



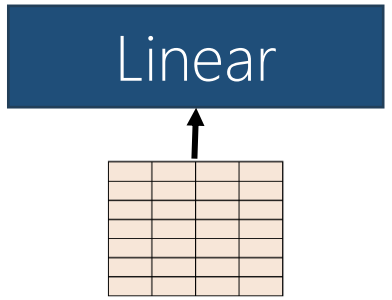
linear layer to convert 7x4 mat back to word

What to do now?

Convert 7x4 to map to vocab

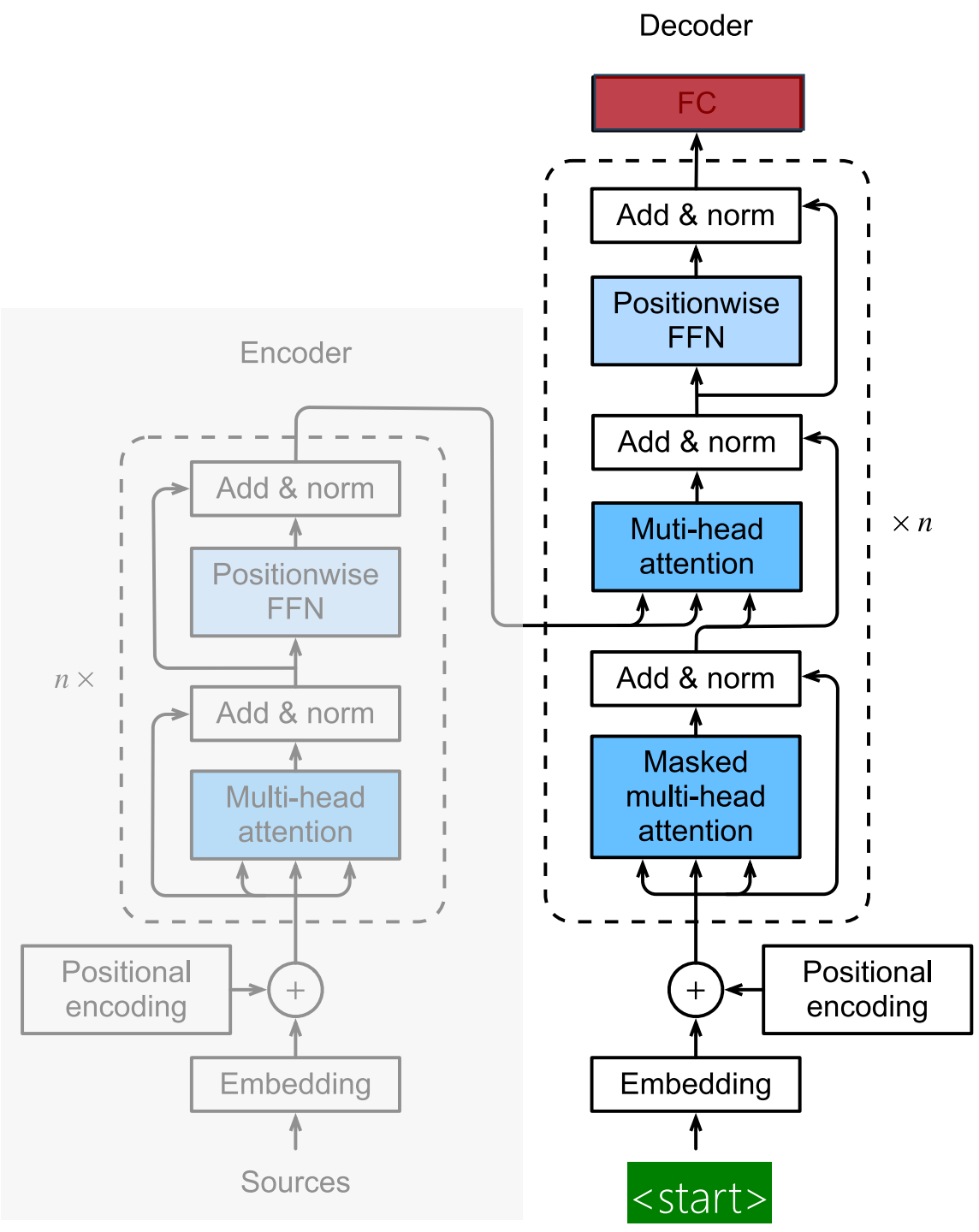


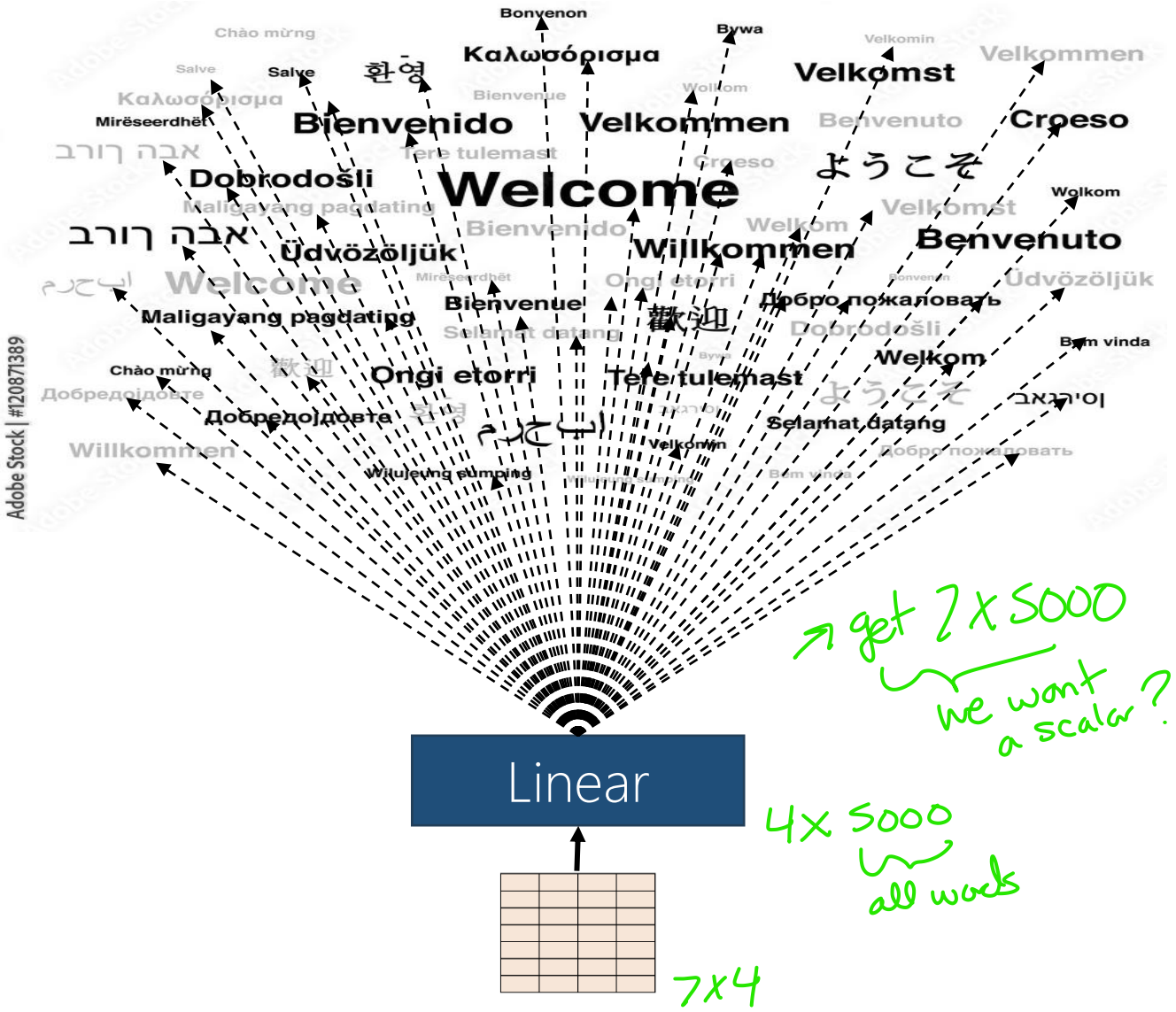
What is best word to predict next in vocab

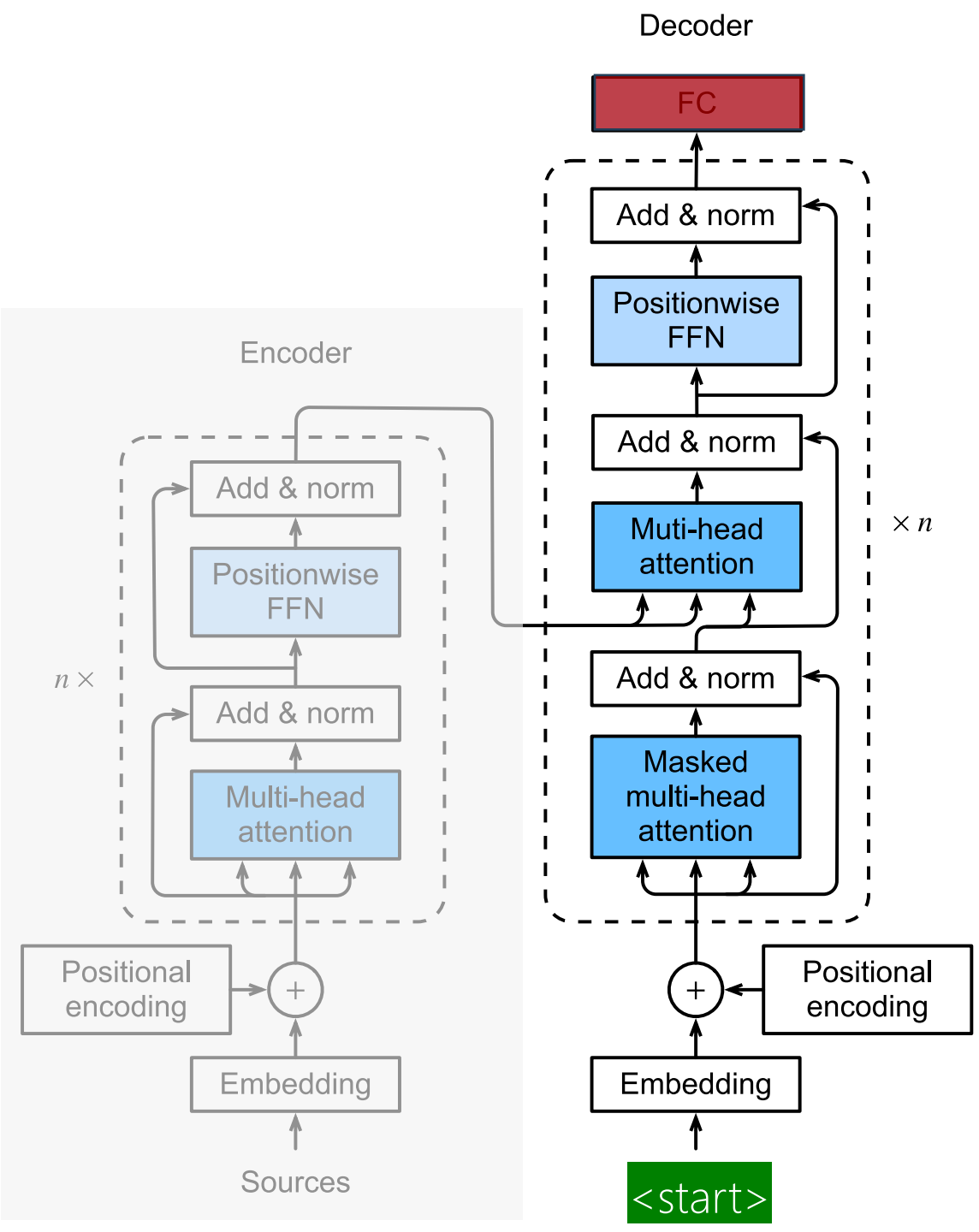


- lin layer dimension 4×5000

7 x 4 .


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Positional encoding
Embedding
<start>
Encoder
n ×
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
Positional encoding
Embedding
Sources


Bienvenido
Welcome
Velkommen
Velkomst
Willkommen
Benvenuto
Croeso
Dobrodošli
Üdvözöljük
Maligayang pagdating
Bienvenue
Ongi etorri
Tere tulemast
Selamat datang
Добро пожаловать
Linear
get 7x5000
we want a scalar?
4x5000
all words
7x4
Adobe Stock | #120871389



Softmax

Linear

get probabilities and select top word but don't have a vector

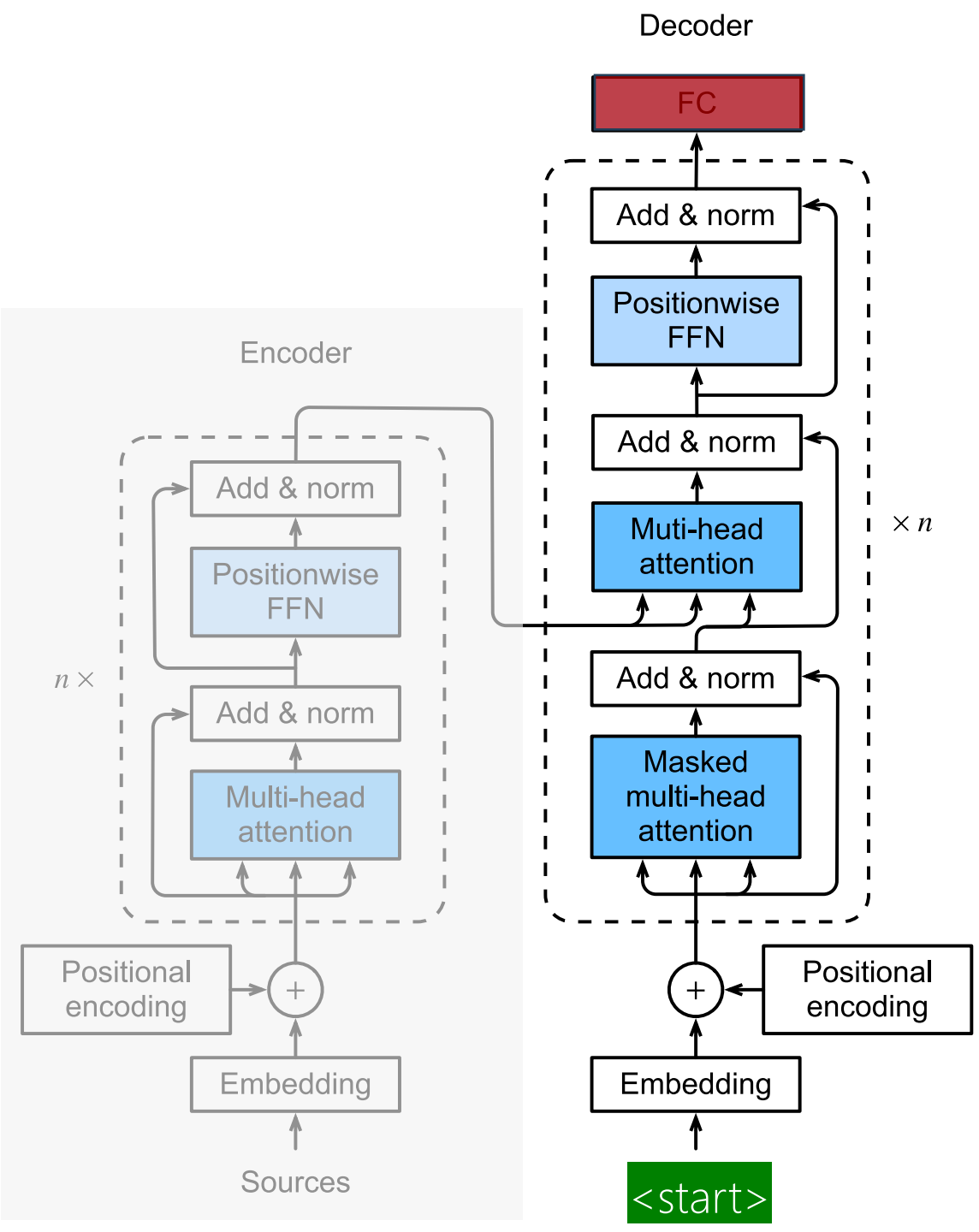
Squishing into a vector happens here

4 x 5000

7 x 5000

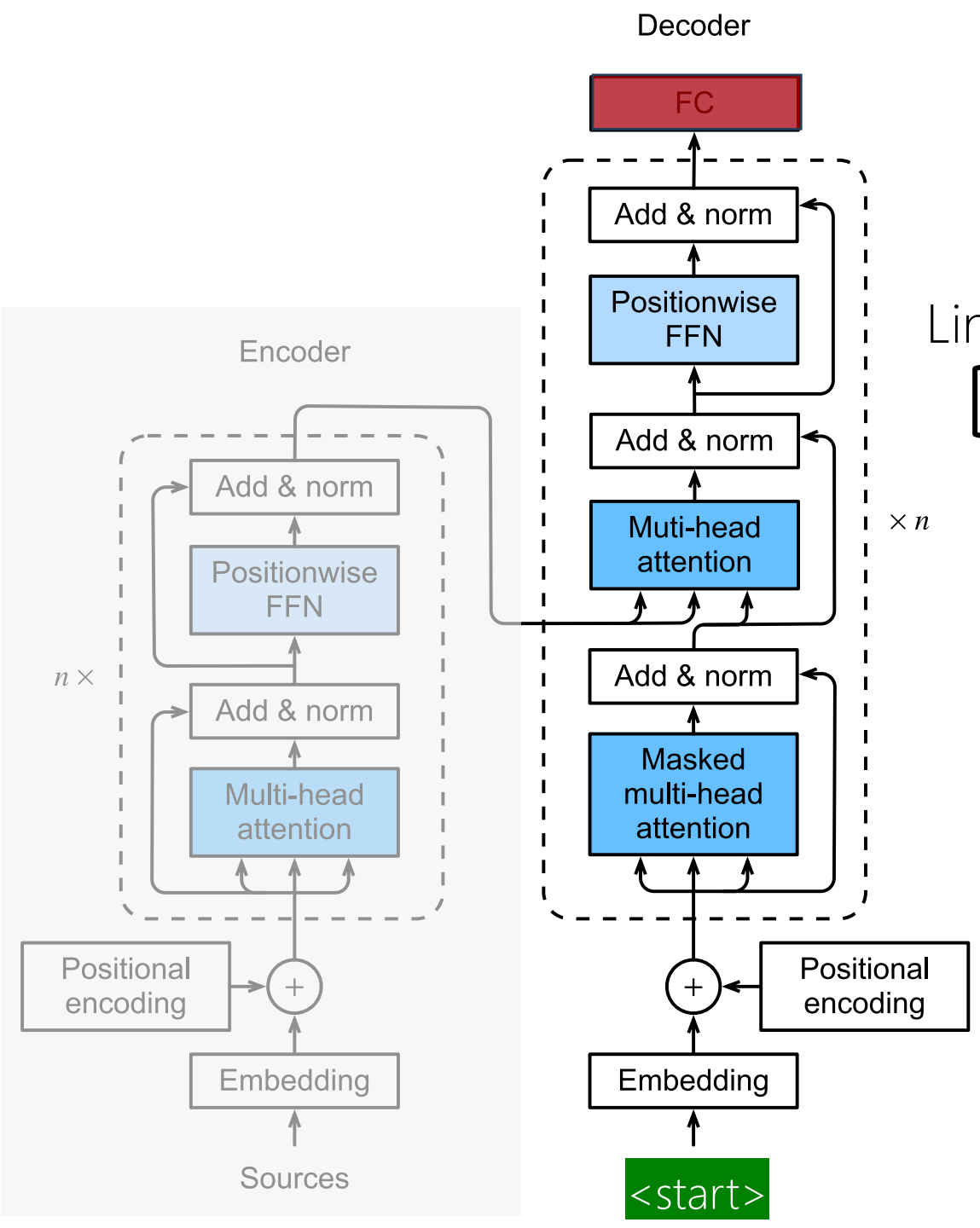
not pad

7 x 4



Wait a minute!!! Is this even possible?

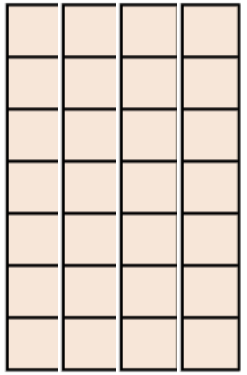
No! Need a vector, have a 7×5000 matrix



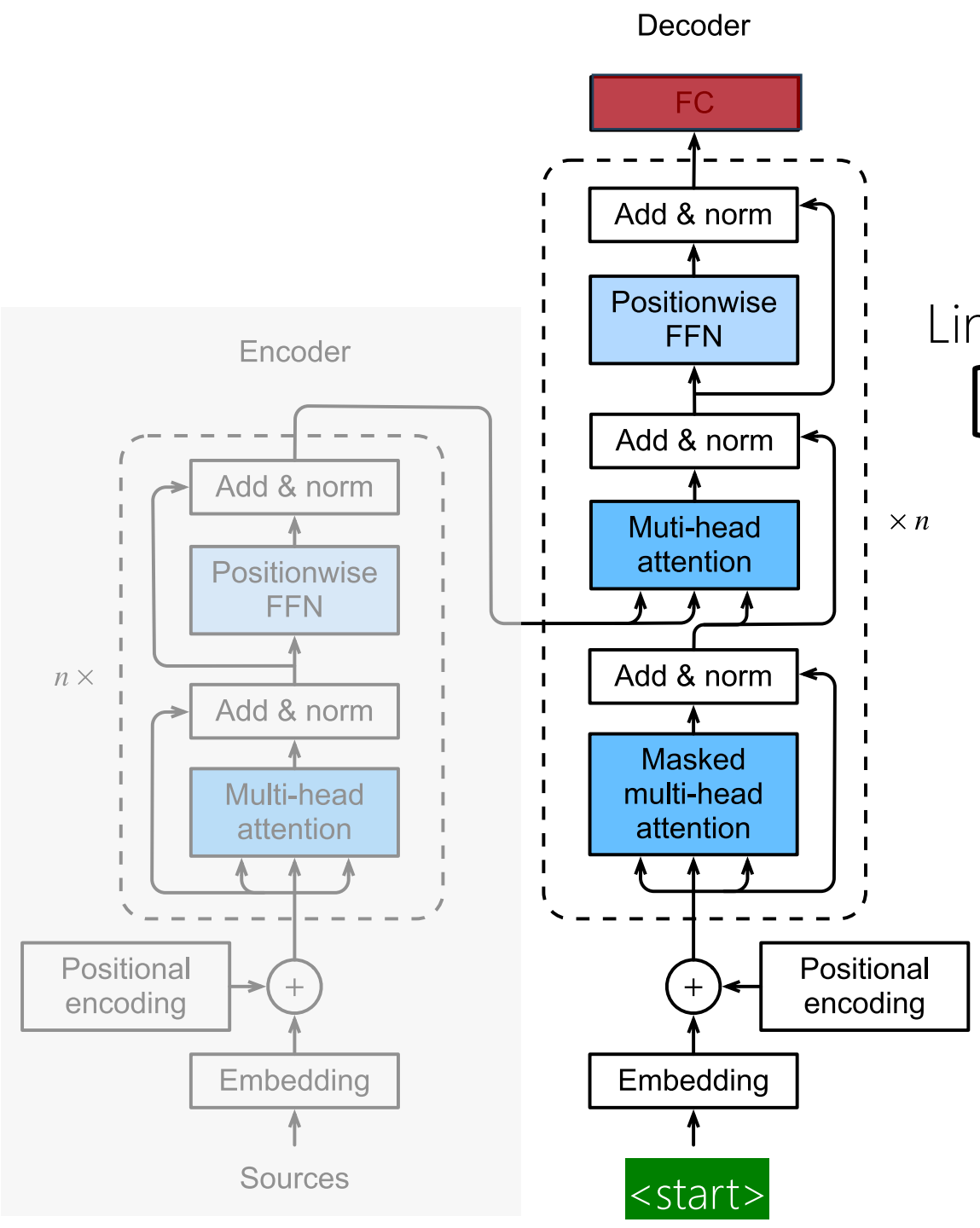
Linear

[0 1 2 501 612 697 ... 891 1111 1200 1893 3019 5001 ... 8539 ... 25213 ... 49999]

5000 vocab



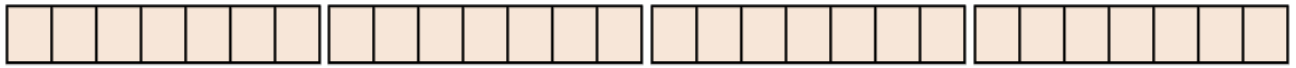
7x4

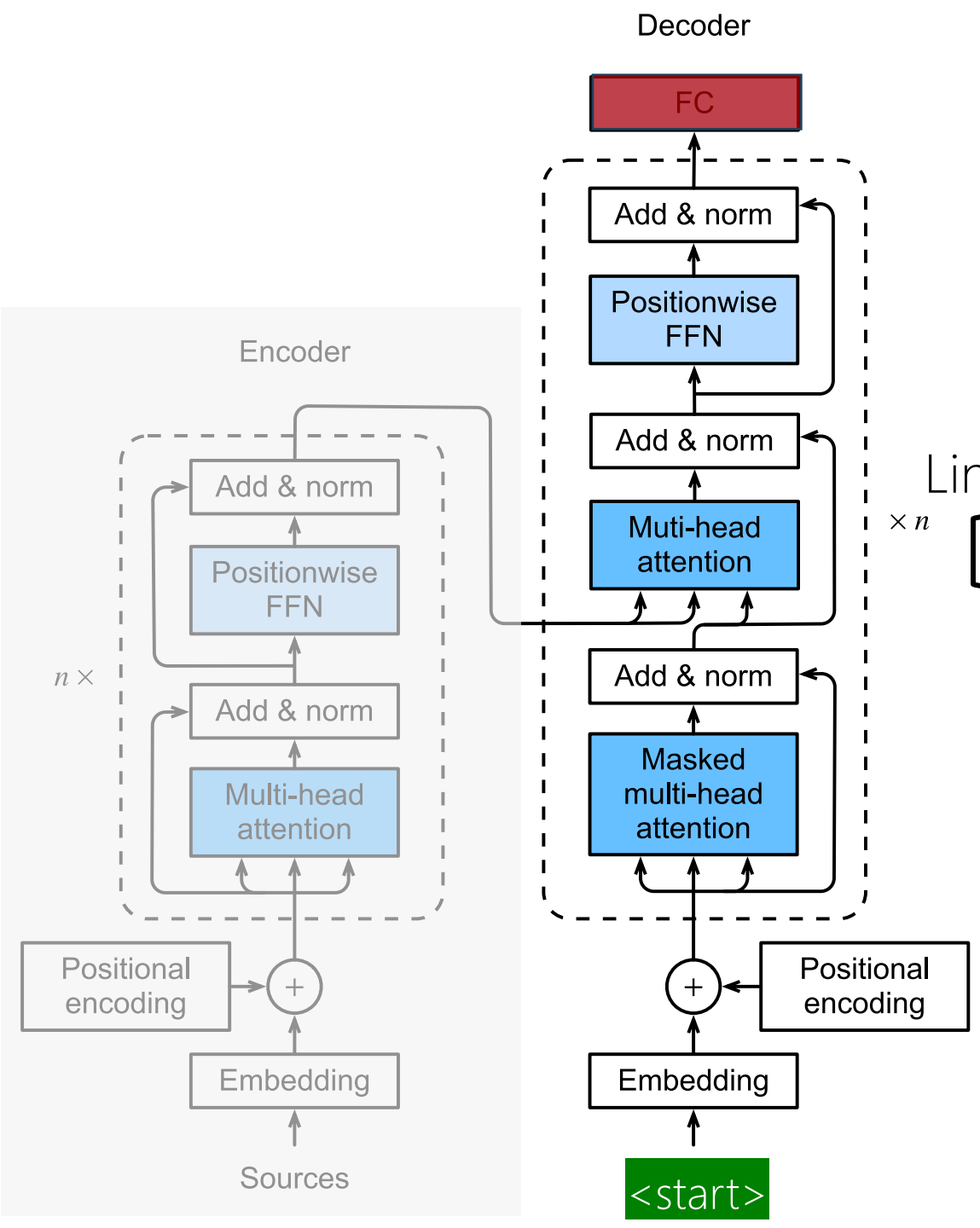


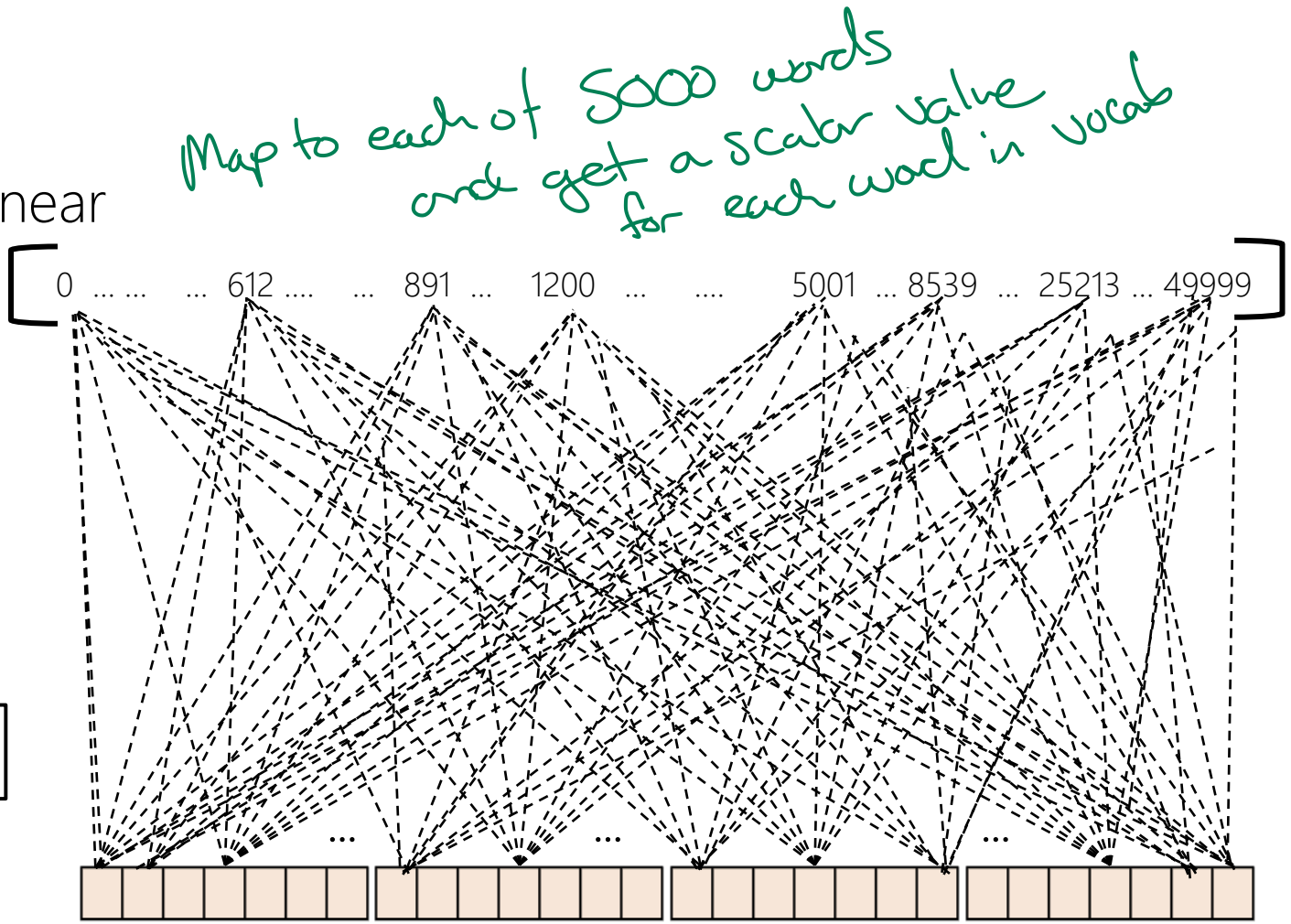
Linear

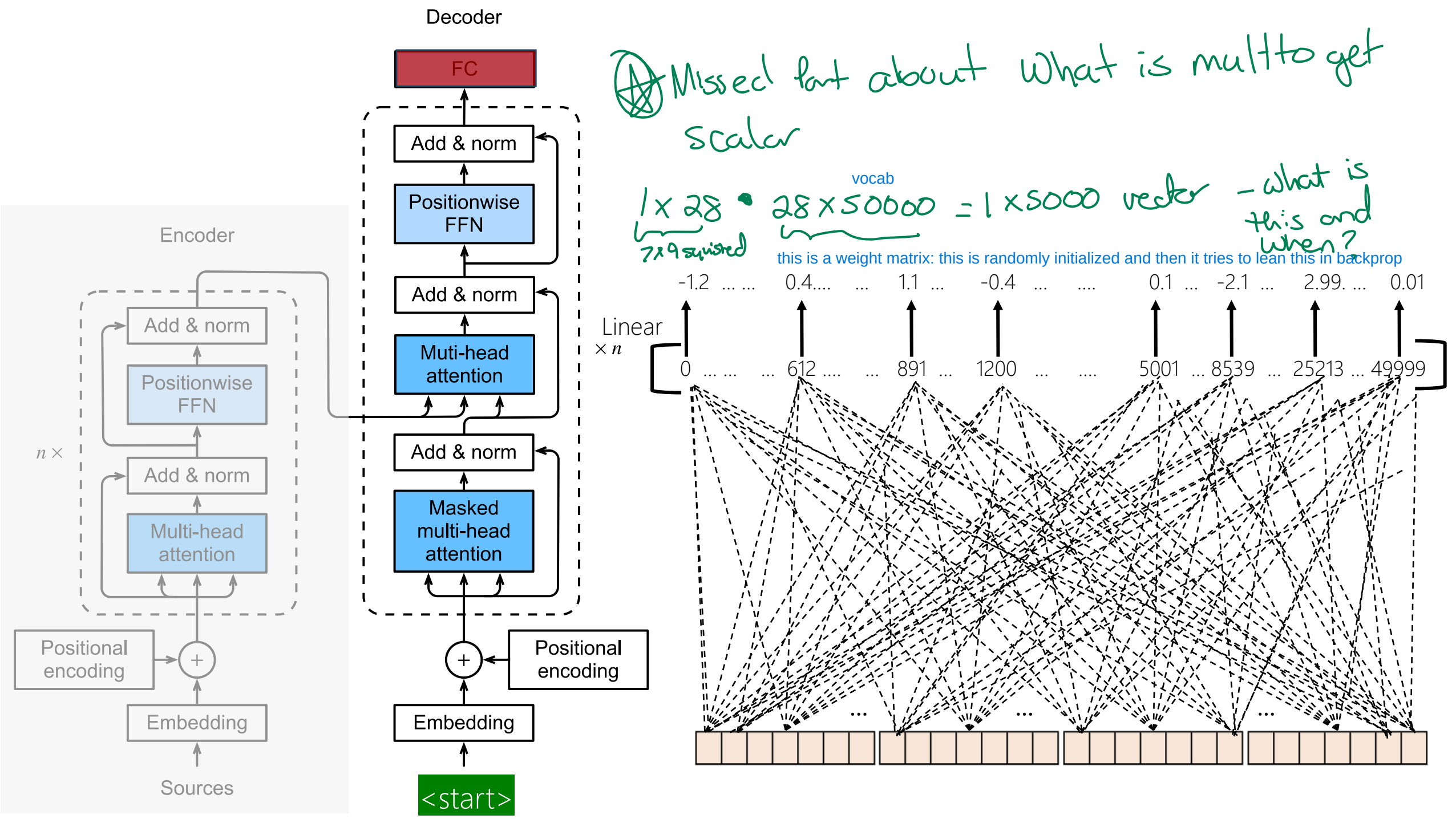
[0 1 2 501 612 697 ... 891 1111 1200 1893 3019 5001 ... 8539 ... 25213 ... 49999]

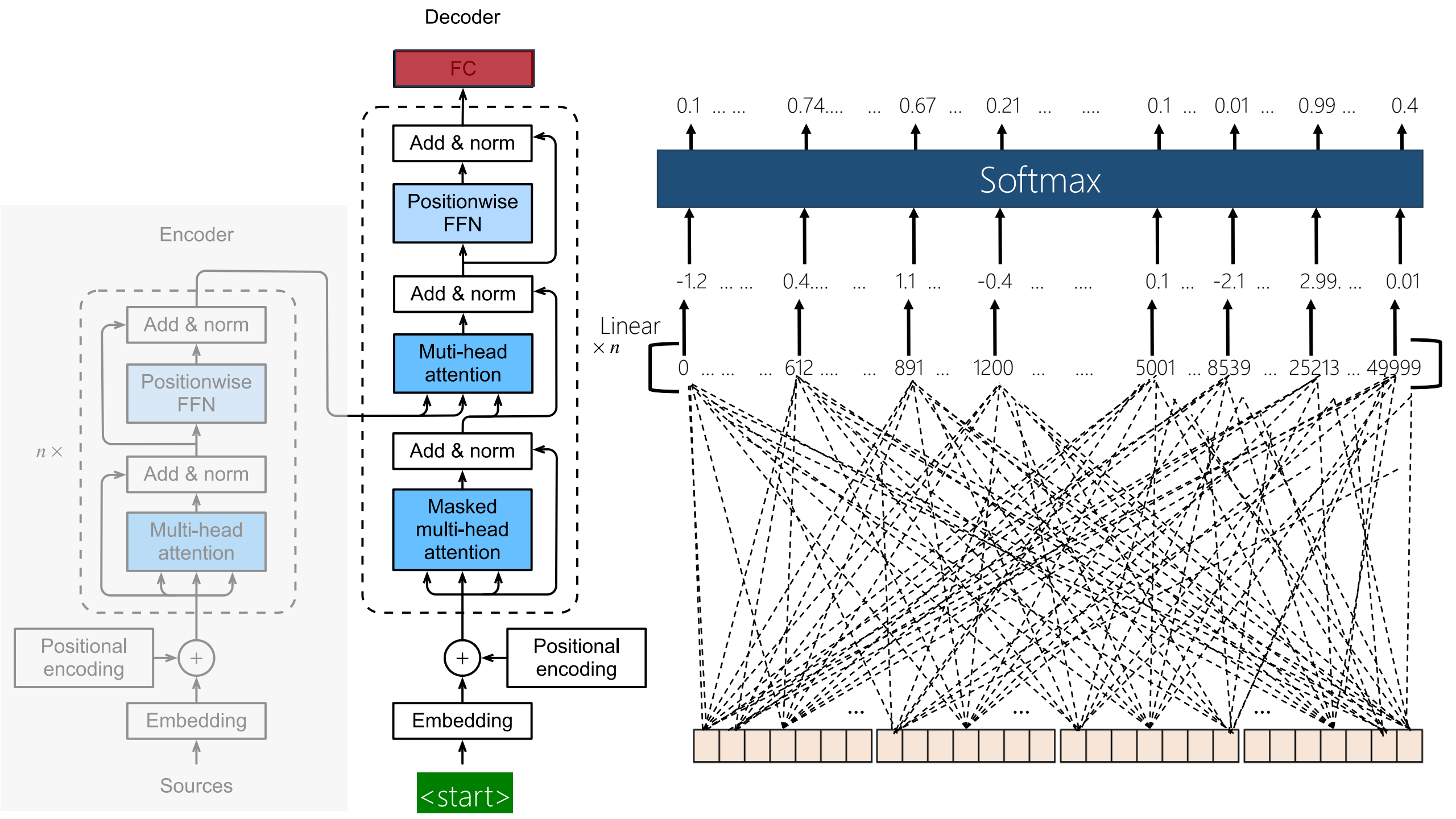
Flatten into a 2§ dim vector ⇒ col-wise (concat reps of each word)

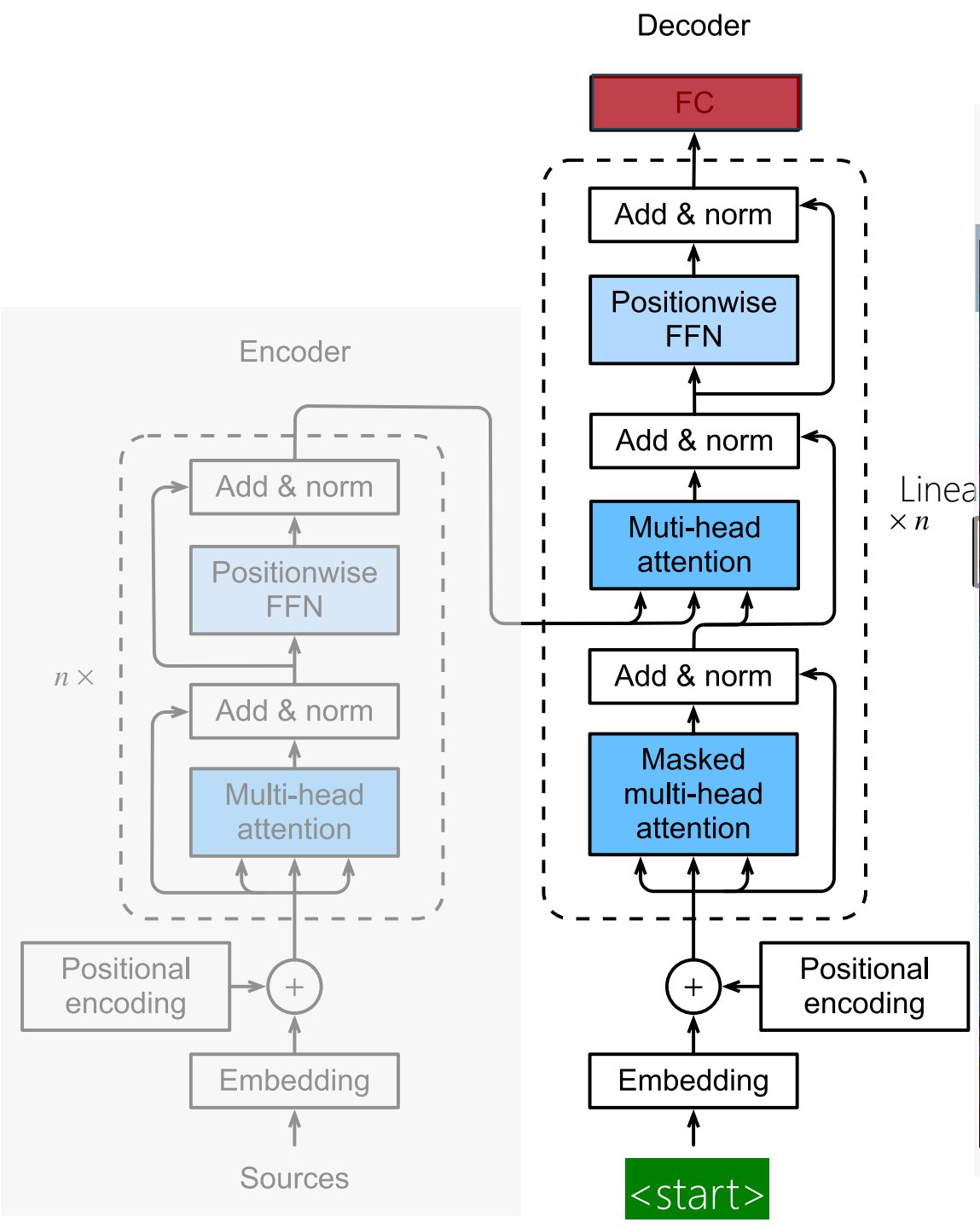



Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Positional encoding
Embedding
<start>
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Embedding
Sources

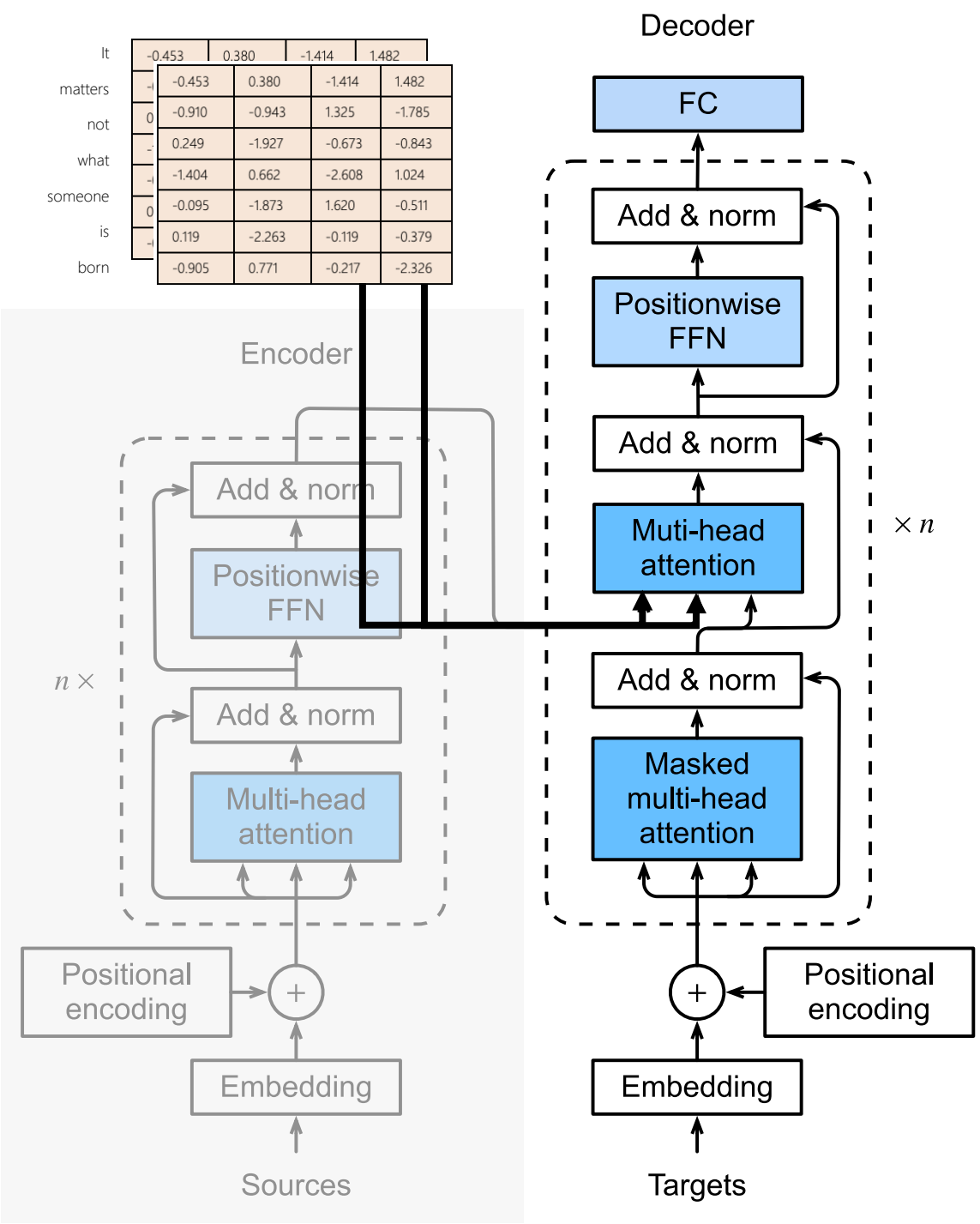

Linear
Map to each of 5000 words and get a scalar value for each word in vocab
0 ... ... 612 .... 891 ... 1200 ... . 5001 ... 8539 ... 25213 ... 49999


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
Positional encoding
Embedding
<start>
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Embedding
Sources
Missed part about What is multto get scalar
1 x 28 squished
28 x 50000 = 1 x 5000 vector
vocab
what is this and when?
this is a weight matrix: this is randomly initialized and then it tries to lean this in backprop
Linear
× n
-1.2 ... ... 0.4.... ... 1.1 ... -0.4 ... . 0.1 ... -2.1 ... 2.99. ... 0.01
0 ... ... 612 .... 891 ... 1200 ... . 5001 ... 8539 ... 25213 ... 49999

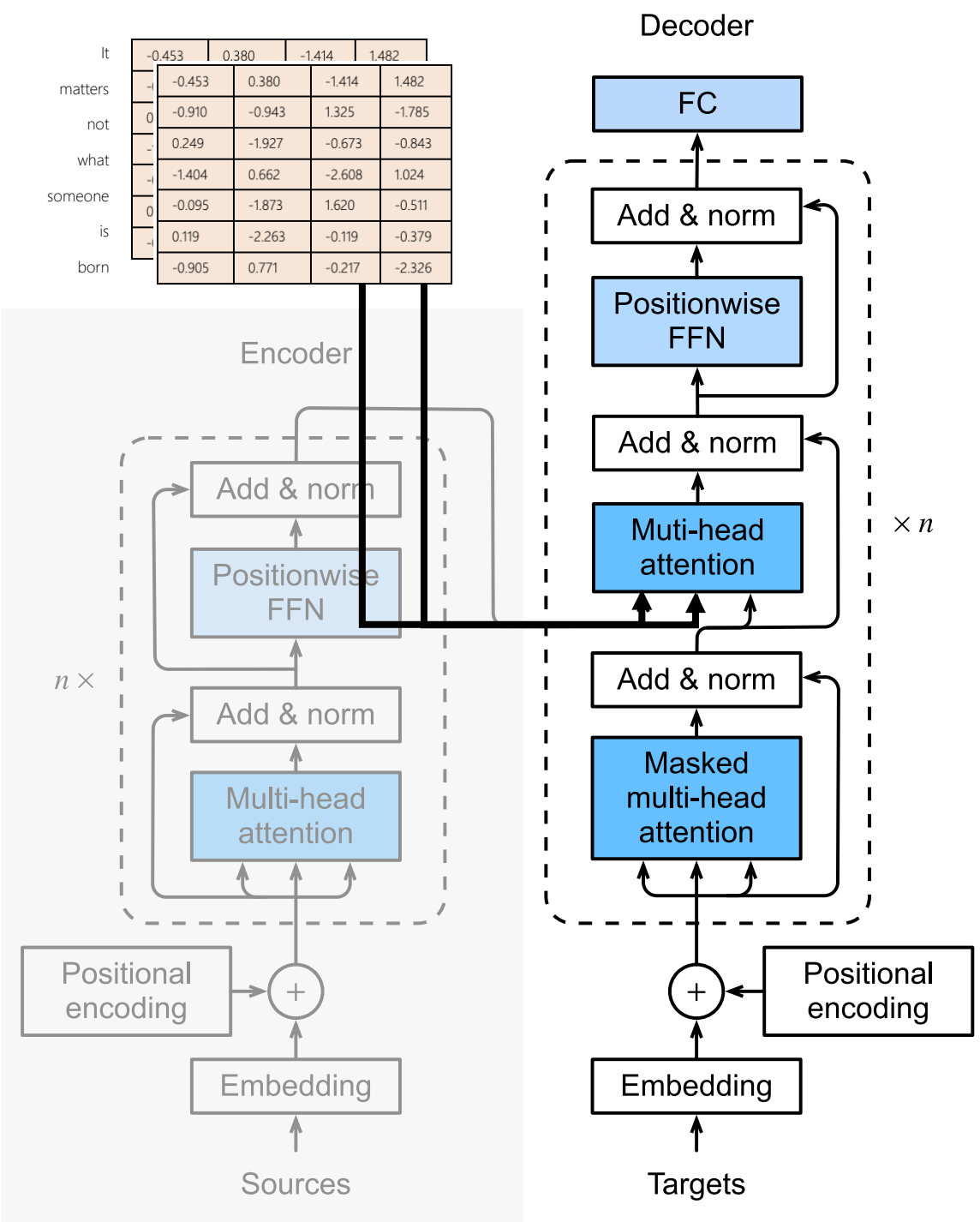

Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
Positional encoding
Embedding
<start>
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Embedding
Sources
Linear
× n
Softmax
0.1 ... ... 0.74.... ... 0.67 ... 0.21 ... . 0.1 ... 0.01 ... 0.99 ... 0.4
-1.2 ... ... 0.4.... ... 1.1 ... -0.4 ... . 0.1 ... -2.1 ... 2.99. ... 0.01
0 ... ... 612 .... 891 ... 1200 ... . 5001 ... 8539 ... 25213 ... 49999


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
Positional encoding
Embedding
<start>
Linea
× n
Encoder
n ×
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
Positional encoding
Embedding
Sources

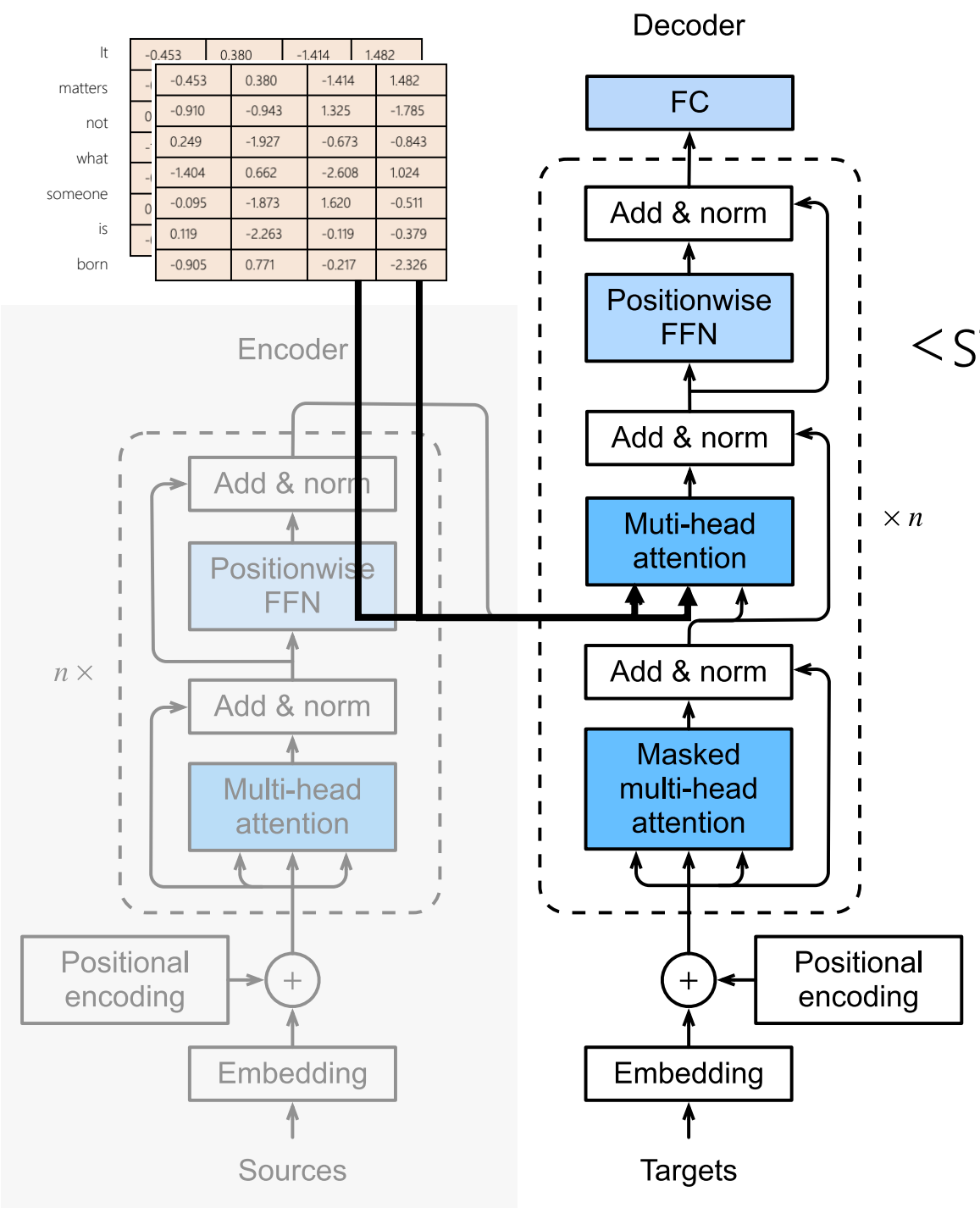

but
0.1 ... ...
0.74....
...
0.67 ...
0.21 ...
....
0.1 ...
0.01 ...
0.99 ...
0.4
WE HAVE A WINNER!


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Positional encoding
Embedding
Targets
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Embedding
Sources
It
matters
not
what
someone
is
born
-0.453 0.380 -1.414 1.482
-0.910 -0.943 1.325 -1.785
0.249 -1.927 -0.673 -0.843
-1.404 0.662 -2.608 1.024
-0.095 -1.873 1.620 -0.511
0.119 -2.263 -0.119 -0.379
-0.905 0.771 -0.217 -2.326

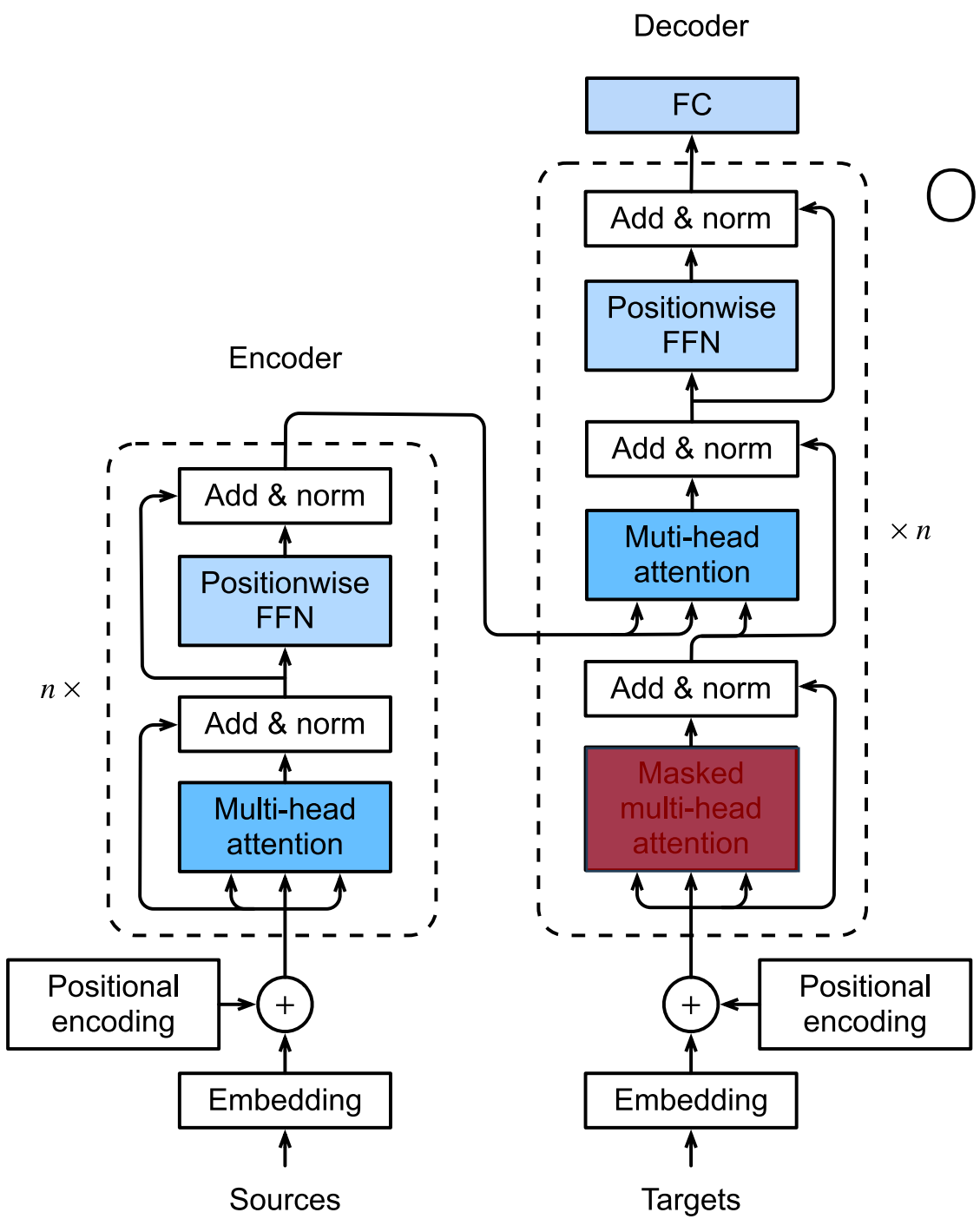
<start>


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Positional encoding
Embedding
Targets
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Embedding
Sources
It
matters
not
what
someone
is
born
-0.453 0.380 -1.414 1.482
-0.910 -0.943 1.325 -1.785
0.249 -1.927 -0.673 -0.843
-1.404 0.662 -2.608 1.024
-0.095 -1.873 1.620 -0.511
0.119 -2.263 -0.119 -0.379
-0.905 0.771 -0.217 -2.326

<start> but



<start> but what they grow to be <end>



Oh wait, we never discussed the masked multi-head attention!!!

Not showing the future and trying to influence the past???

The Dialogue Completer Task

Input Dialogue	Dialogue Completion
It is our choices, Harry, that show what we truly are,	?
If you want to know what a man's like, take a good look at	?
It matters not what someone is born,	?

The Dialogue Completer Task

Input Dialogue

It is our choices, Harry, that show what we truly are,

If you want to know what a man's like, take a good look at

It matters not what someone is born,

Dialogue Completion

<start> far more than our abilities <end>

<start> how he treats his inferiors, not his equals <end>

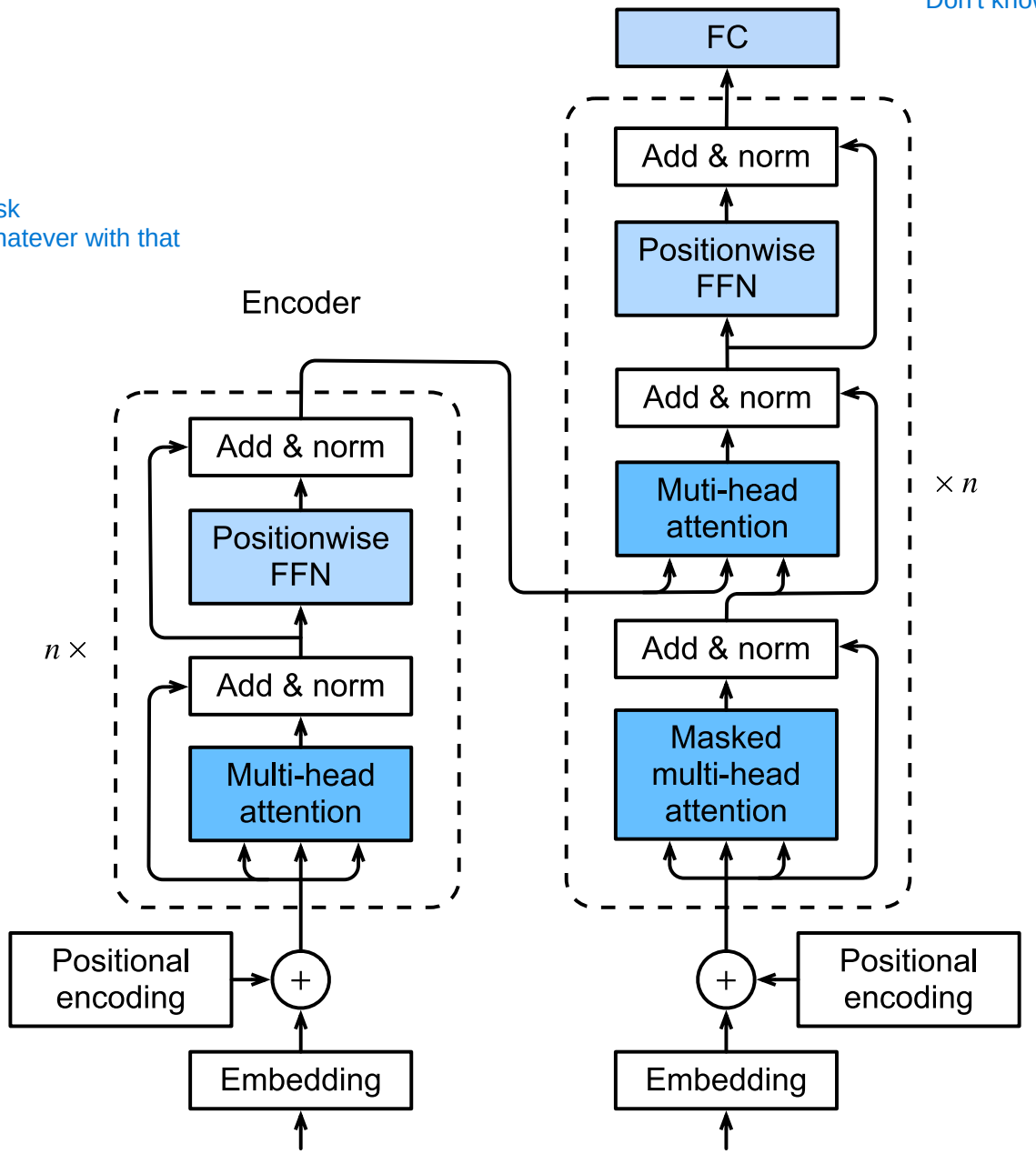
<start> but what they grow to be <end>

Training

Let the model just learn from the data not the task
Analogy: give toys to a child and child can do whatever with that

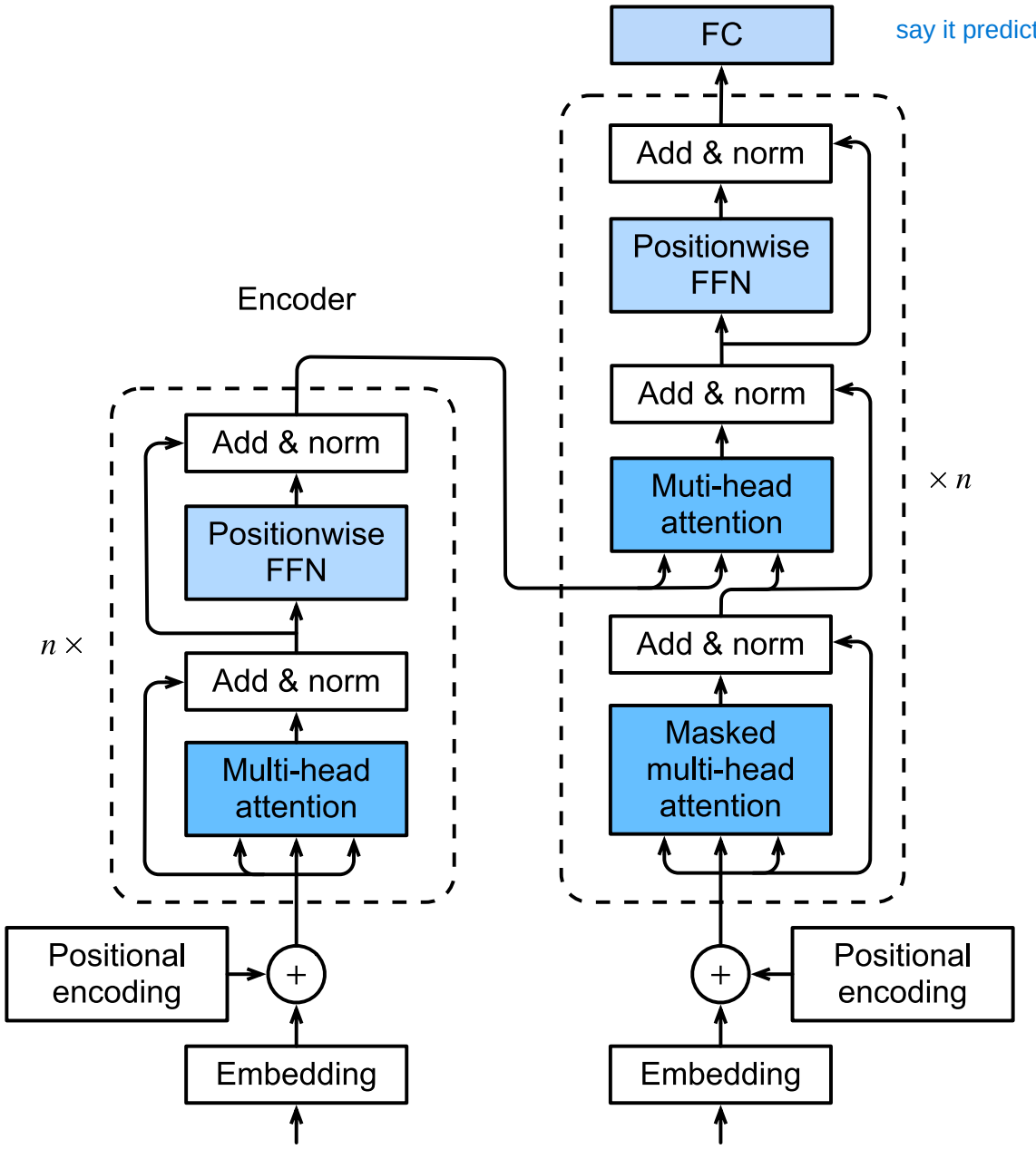
Decoder <start> MASK MASK MASK MASK MASK MASK<end>

Don't know what want to generate=mask



It matters not what someone is born,

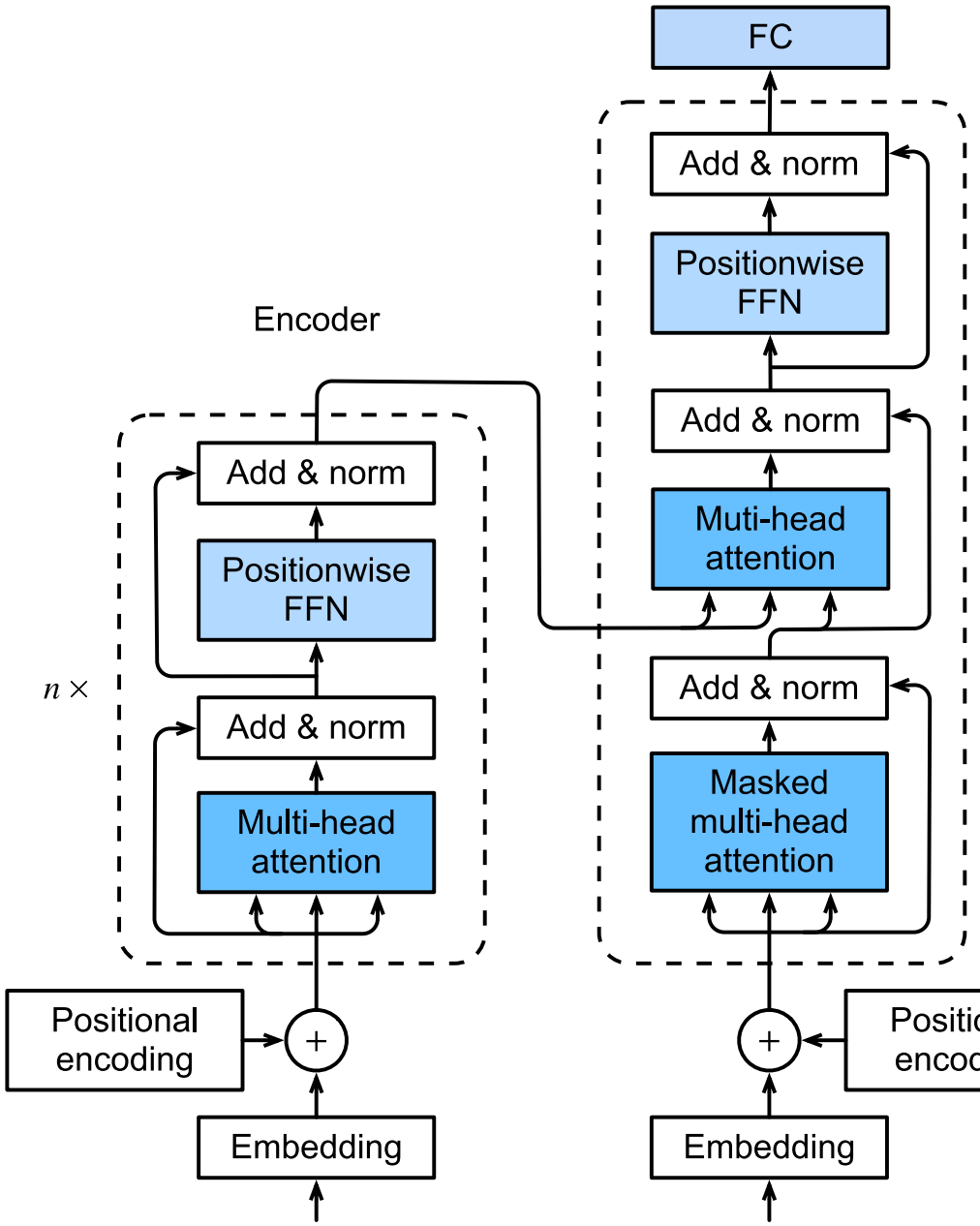
Training


Decoder
Chirag MASK MASK MASK MASK MASK<end>
FC
say it predicted his name as the first thing
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
Positional encoding
Embedding
Embedding
It matters not what someone is born,
<start>

Training

LLM is not generative, it is predictive, just trying to predict the next word

Every word is a class, a unique token



$Loss(P('Chirag'), P('but'))$

prob of predicting word chirag

pass ground truth and get that words prob,

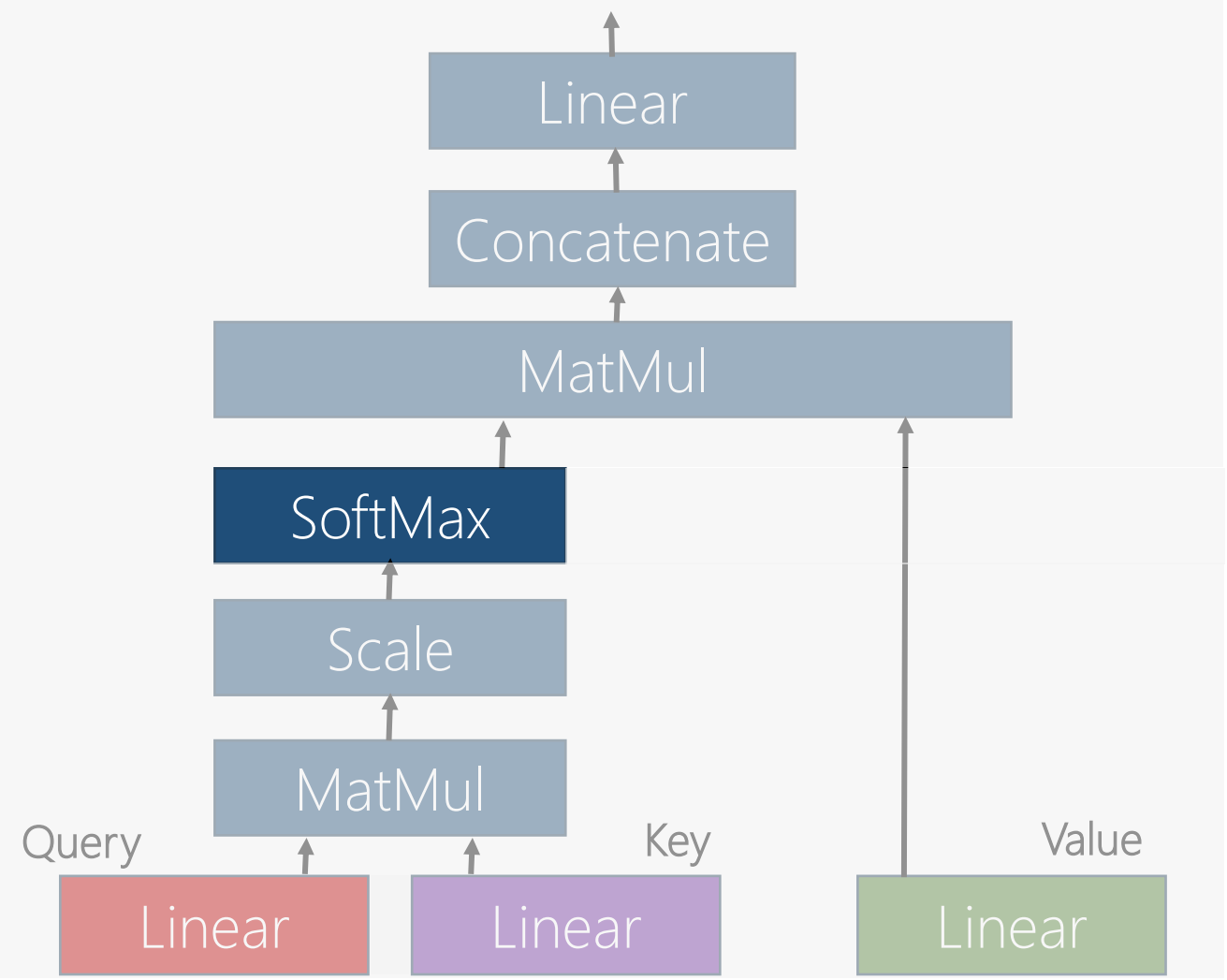
calc the cross entropy loss and then use that to backprop

Ground Truth:

<start> but what they grow to be <end>

take prob of second word, calc loss, back prop, etc
This is why the training is so slow

Masked Attention



	<start>	but	what	they	grow	to	be	<end>
<start>	92	35	54	11	39	91	58	7
but	20	21	67	47	13	61	62	3
what	94	54	76	85	39	49	0	58
they	51	53	72	69	97	46	94	32
grow	8	39	22	85	66	95	7	27
to	1	77	5	73	41	20	50	36
be	21	90	3	7	92	69	56	97
<end>	91	68	0	56	77	59	81	28

Self attention was taking the few same word and

Masked Attention

If just tell the

soft max includes e^x so soft max of that matrix is 1 if 0 and 0 if -inf

	<start>	but	what	they	grow	to	be	<end>
<start>	92	35	54	11	39	91	58	7
but	20	21	67	47	13	61	62	3
what	94	54	76	85	39	49	0	58
they	51	53	72	69	97	46	94	32
grow	8	39	22	85	66	95	7	27
to	1	77	5	73	41	20	50	36
be	21	90	3	7	92	69	56	97
<end>	91	68	0	56	77	59	81	28

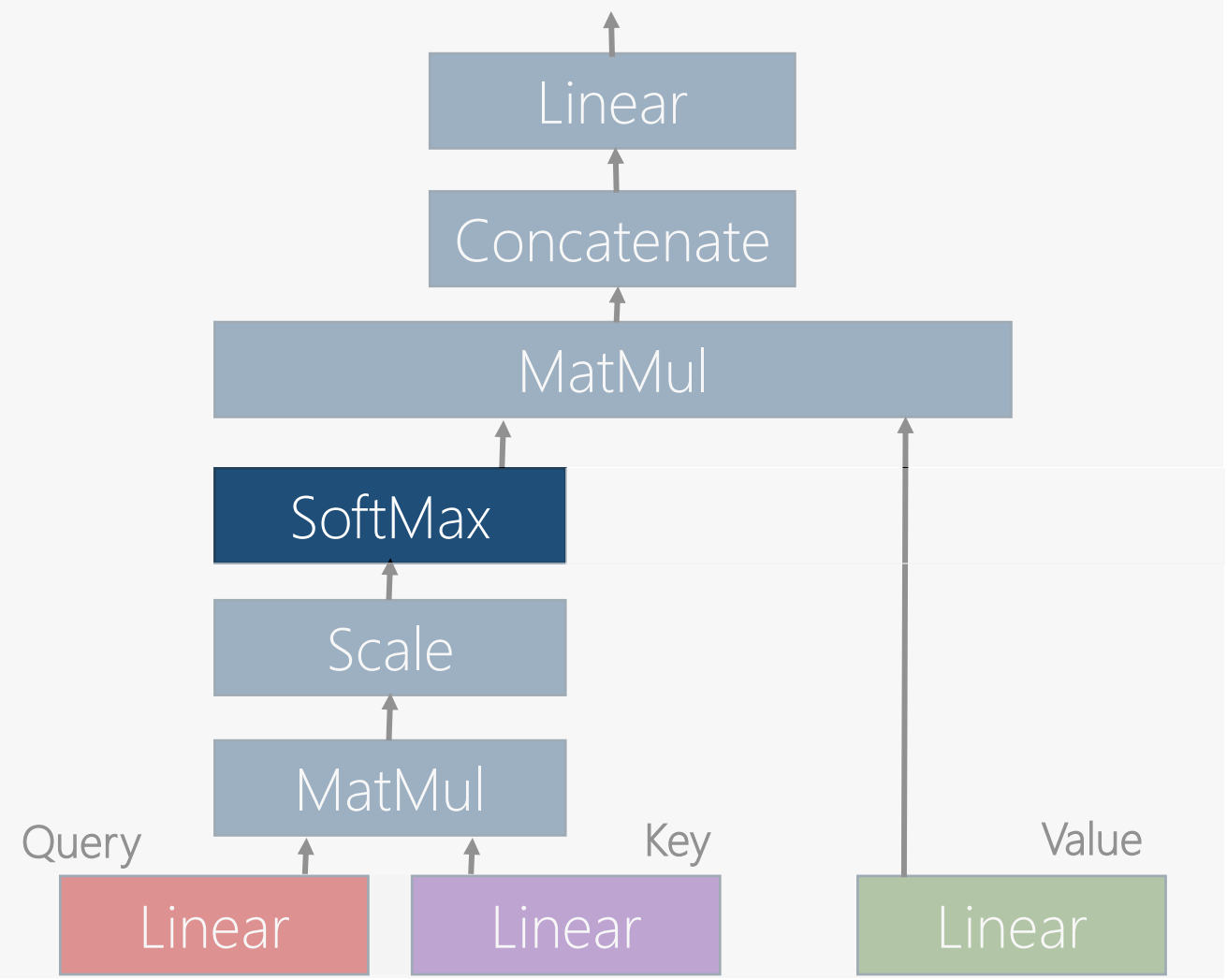
Attention Filter

+

	<start>	but	what	they	grow	to	be	<end>
<start>	0	-inf	-inf	-inf	-inf	-inf	-inf	-inf
but	0	0	-inf	-inf	-inf	-inf	-inf	-inf
what	0	0	0	-inf	-inf	-inf	-inf	-inf
they	0	0	0	0	-inf	-inf	-inf	-inf
grow	0	0	0	0	0	-inf	-inf	-inf
to	0	0	0	0	0	0	-inf	-inf
be	0	0	0	0	0	0	0	-inf
<end>	0	0	0	0	0	0	0	0

Mask Filter

-inf is no information from other words that they want to learn
like word but is not getting info except from the start token and itself, not giving it any future info, just itself (or the previous words??????)



Masked Attention

	<start>	but	what	they	grow	to	be	<end>
<start>	1	0	0	0	0	0	0	0
but	0.27	0.73	0	0	0	0	0	0
what	1	0	0	0	0	0	0	0
they	0	0	0.95	0.05	0	0	0	0
grow	0	0	0	1	0	0	0	0
to	0	0.98	0	0.02	0	0	0	0
be	0	0.12	0	0	0.88	0	0	0
<end>	1	0	0	0	0	0	0	0

Masked Attention Filter

with the soft max,

THE pt is to make sure the training happens correctly

Next Week!!

- Different Training Stages of LLMs