Lecture 15: Transformer-based Encoders

Claire Cardie, Tanya Goyal

CS 4740 (and crosslists): Introduction to Natural Language Processing

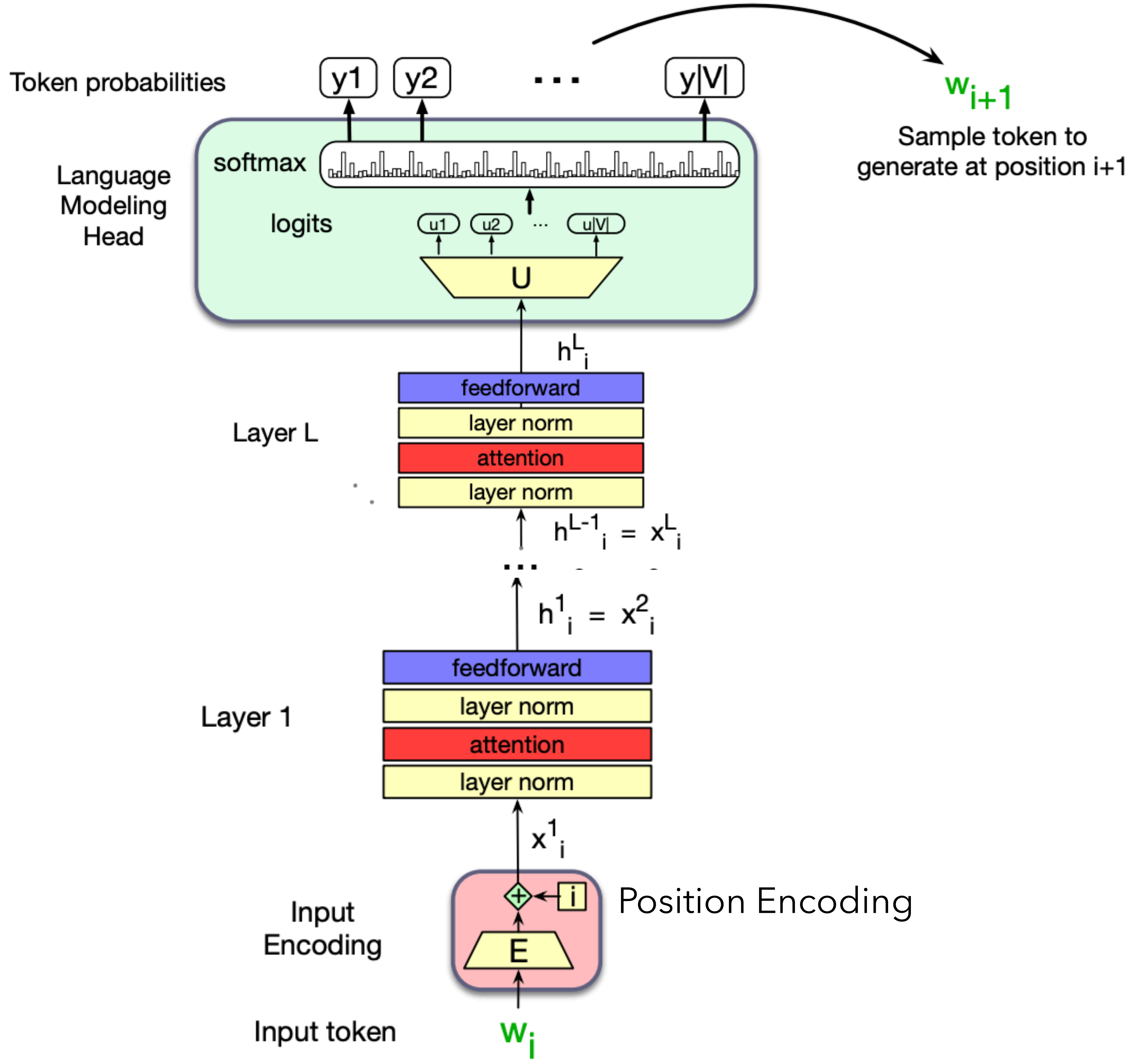
Announcements

- HW3 will be released on Wednesday.

Today

- Recap: Decoder-only transformer models
- Encoder-decoder transformer models
- Encoder-only models (BERT)

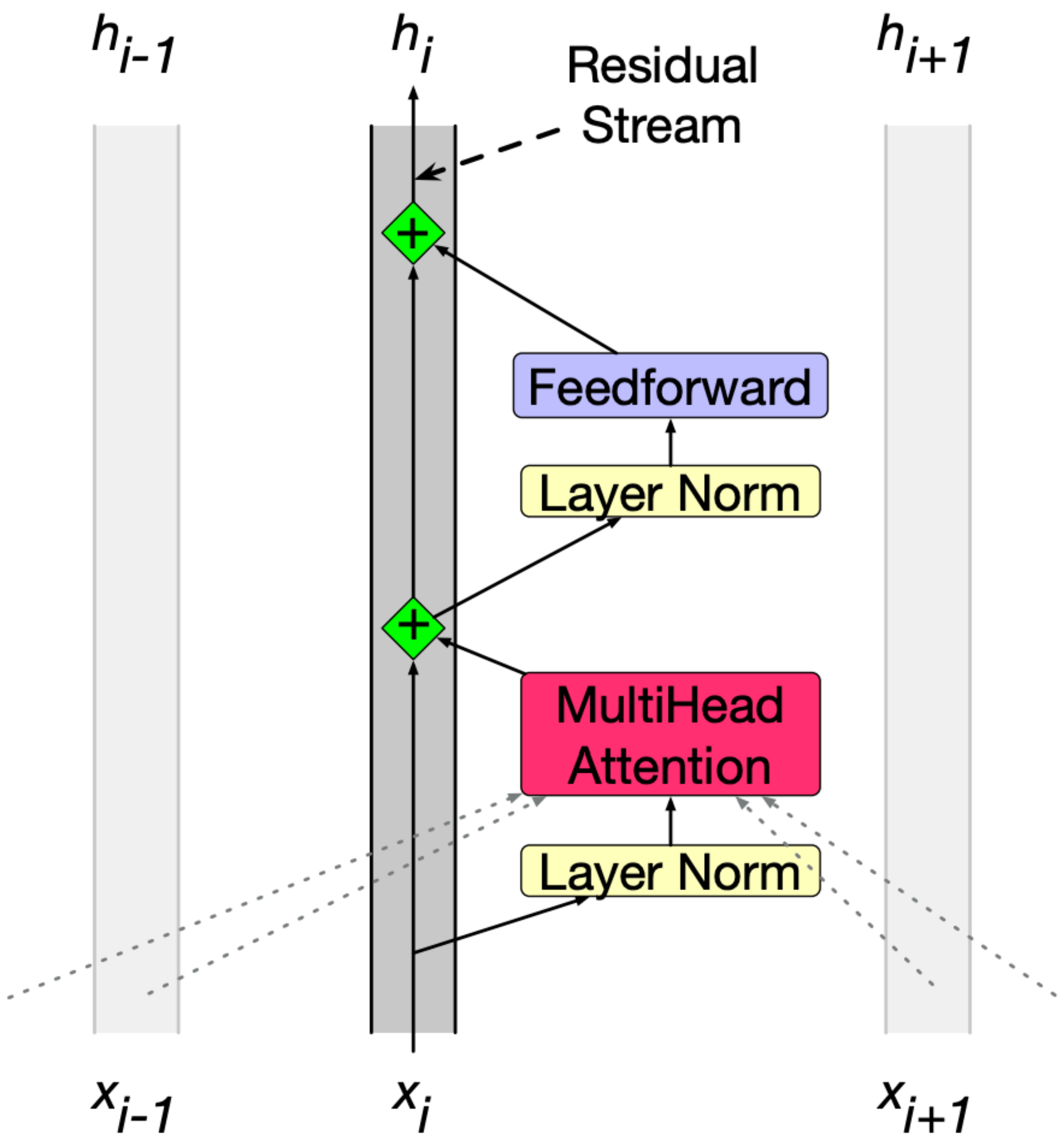
Recap Decoder-only Transformer



- Main components of a transformer model
  - **(Multi-head) Attention**
  - Feed forward
  - Layer Norm

  - Position Encoding

Recap: Residual Stream view



…

Input x_i at time step i

$$t_i^1 = \text{LayerNorm}(x_i)$$

$$t_i^2 = \text{MultiHead-Attention}(t_i^1, [t_1^1, t_2^1, \ldots t_N^1])$$

$$t_i^3 = t_i^2 + x_i$$

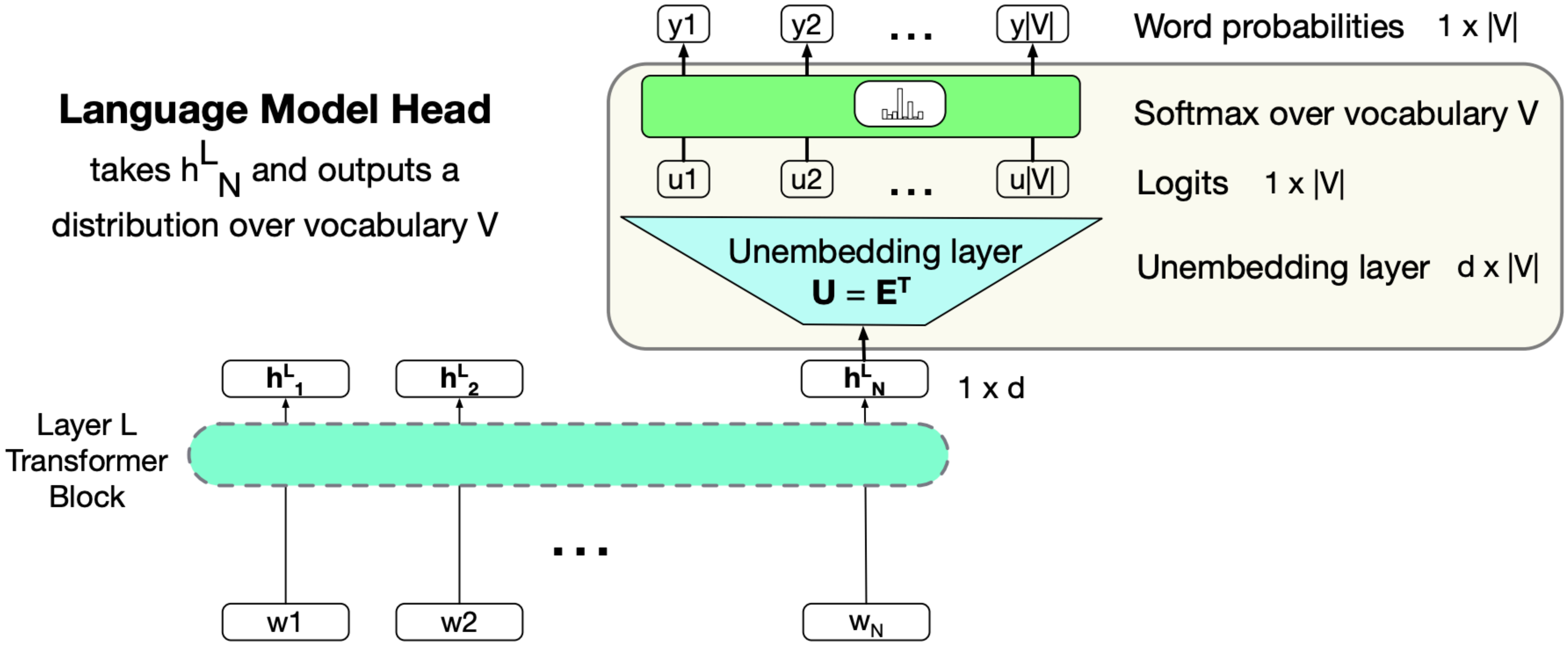
$$t_i^4 = \text{LayerNorm}(t_i^3)$$

$$t_i^5 = \text{FFNN}(t_i^4)$$

$$h_i = t_i^5 + t_i^3$$

Recap: Output Layer

- Final output: probability distribution over the vocabulary.
- Training objective: Predict the next token, given preceding tokens.



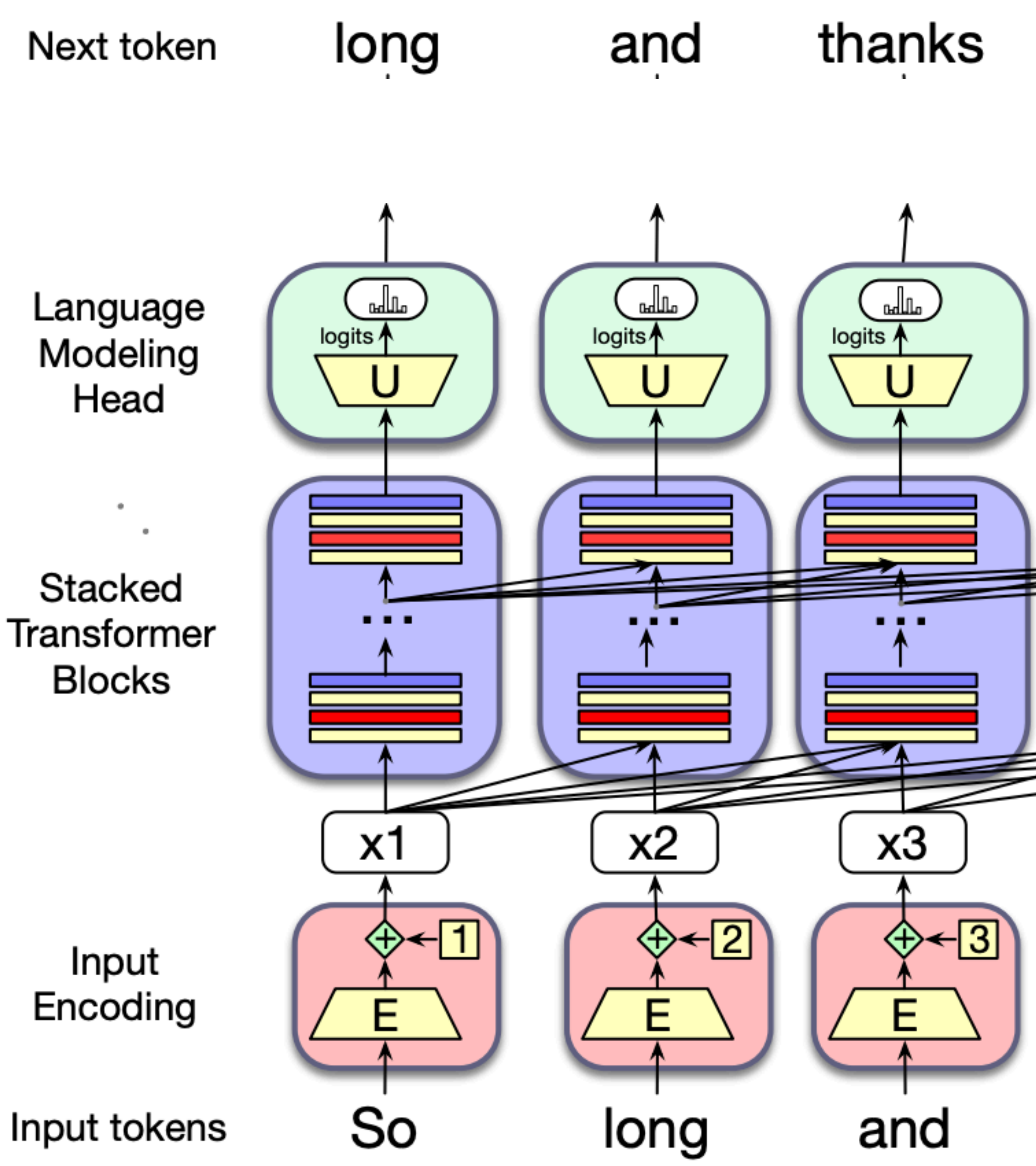
Recap: Language Modeling with Decoder-only Transformers

- Q: How should we model loss?
  - Cross Entropy Loss

$$L_{CE} = -\sum_{w \in V} \mathbf{y}_t[w] \, \log \hat{\mathbf{y}}_t[w]$$

  - Simplifies to:

$$L_{CE} = -\log \hat{\mathbf{y}}_t[w_{t+1}]$$



Large Language Models

- Decoder-only transformer models allows us to model the task of language modeling. Why should we care about language modeling?
- Many practical tasks in NLP can be cast as next token prediction.

Sentiment Analysis:

The sentiment of the sentence “I like Jackie Chan" is:

P(positive|The sentiment of the sentence “I like Jackie Chan" is:)
P(negative|The sentiment of the sentence “I like Jackie Chan" is:)

Large Language Models

- Decoder-only transformer models allows us to model the task of language modeling. Why should we care about language modeling?
- Many practical tasks in NLP can be cast as next token prediction.

Question Answering:

Q: Who wrote the book “The Origin of Species"? A:

P(w | Q: Who wrote the book “The Origin of Species"? A:)

Large Language Models

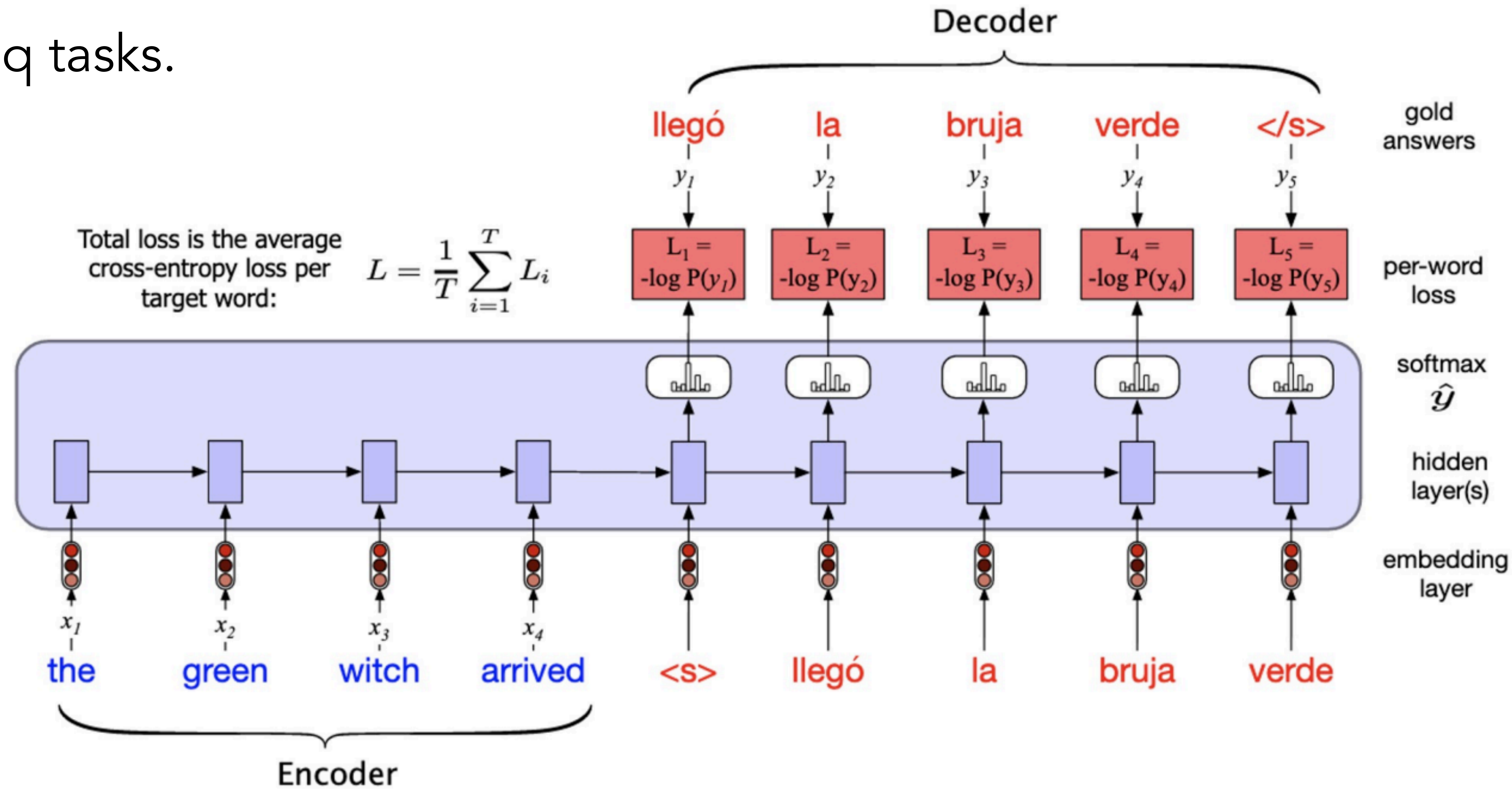
- Decoder-only transformer models allows us to model the task of language modeling. Why should we care about language modeling?
- Many practical tasks in NLP can be cast as next token prediction.

Language models need to be very powerful perform well at all these tasks!

- **Very** deep network
- Train on a lot of data
    - E.g. GPT-3 model (released in 2020) trained on 300B tokens, LLaMA-3 model trained on 15T tokens.

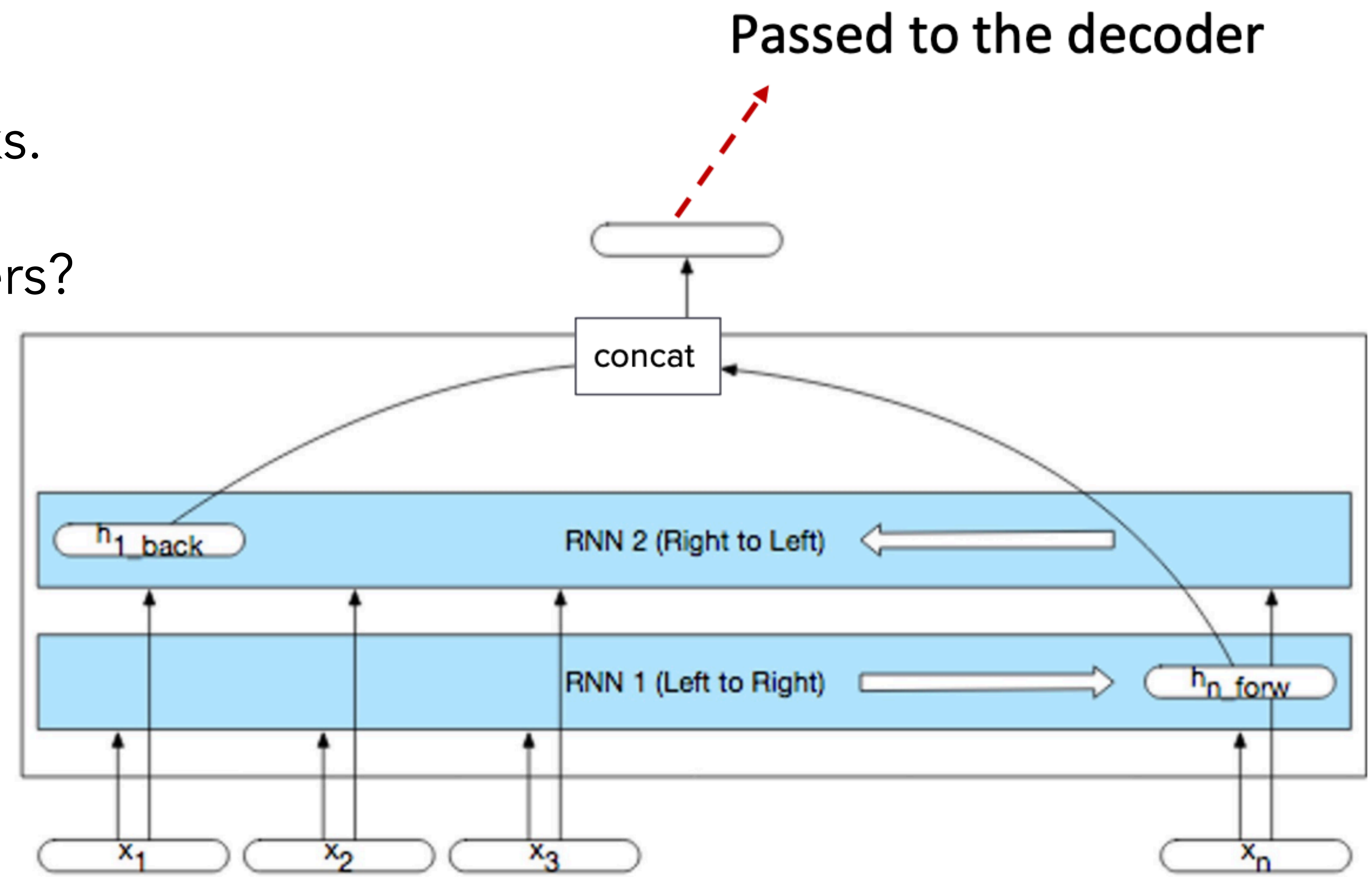
Encoder-Decoder Architecture

- Recall RNNs.
- Useful for seq2seq tasks.

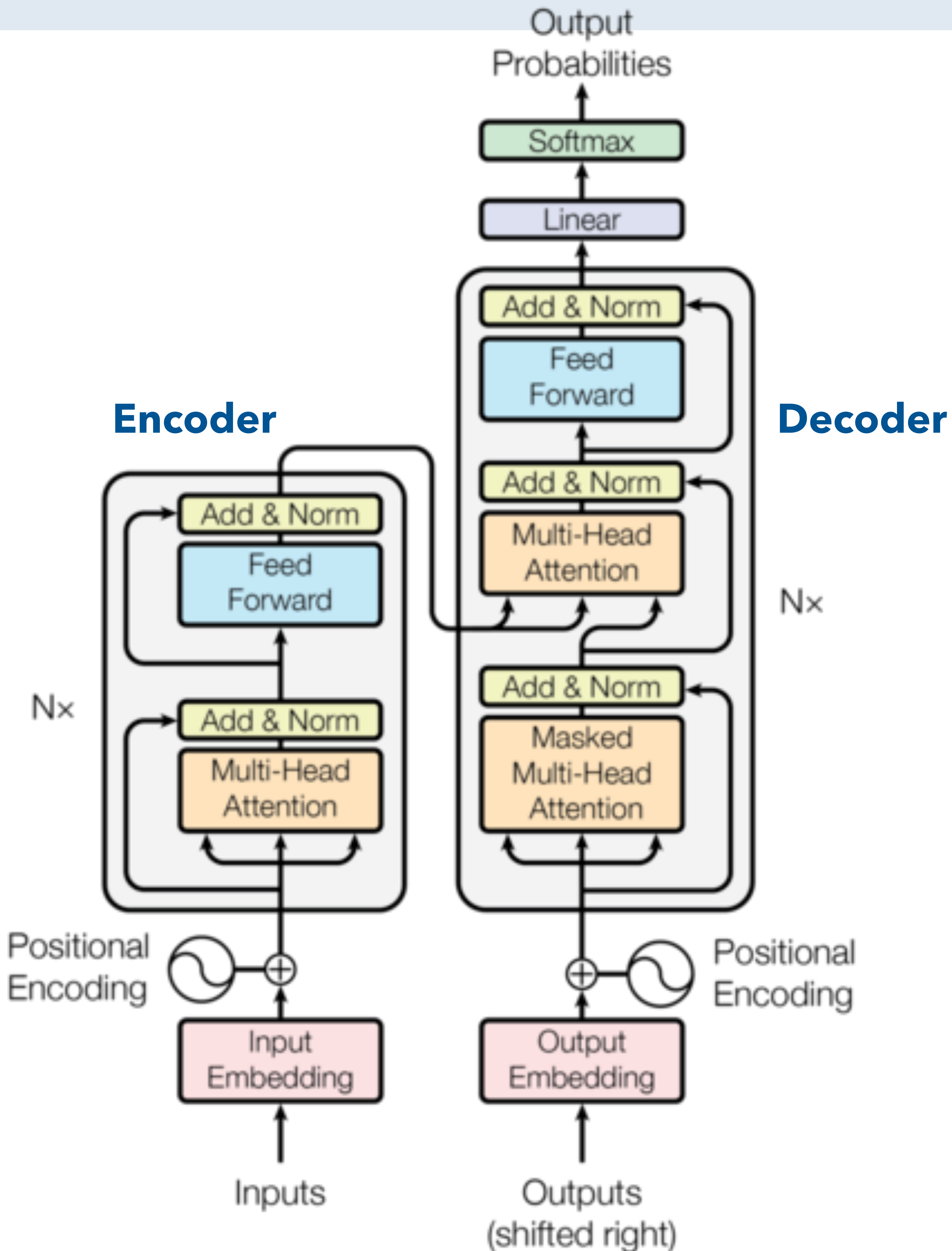


Encoder-Decoder Architecture

- Recall RNNs.
- Useful for seq2seq tasks.
- What about transformers?

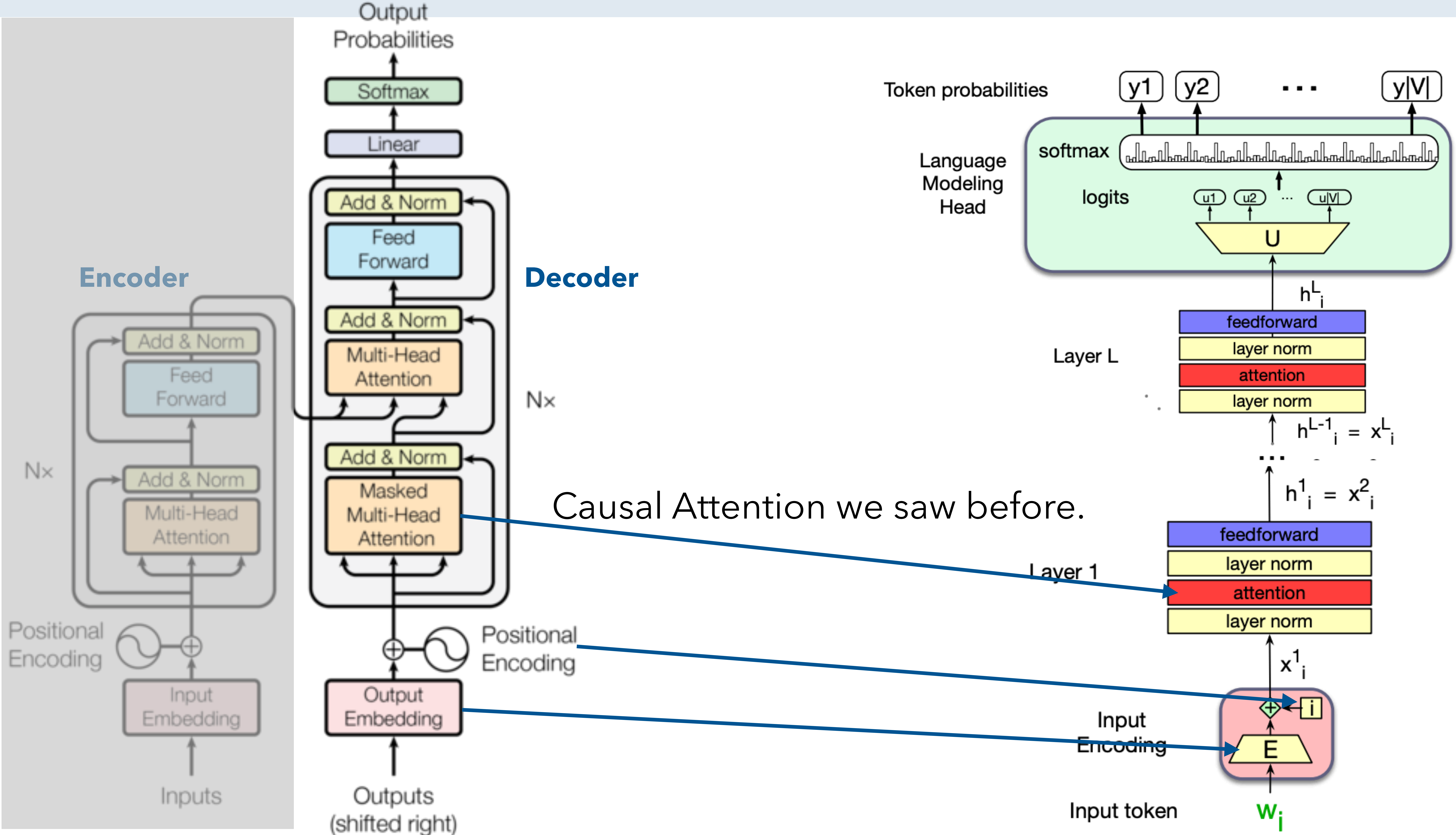


Encoder-Decoder Architecture

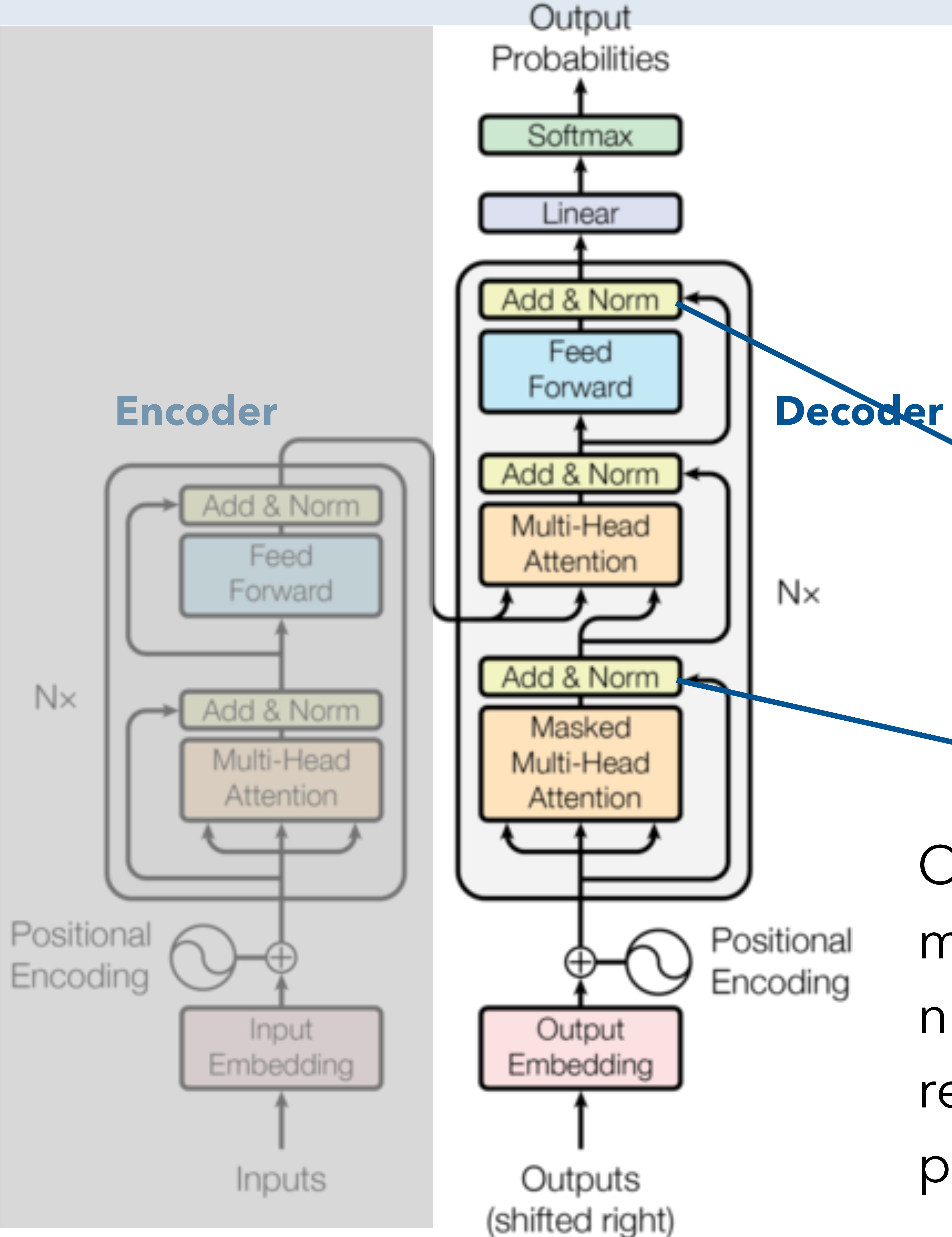


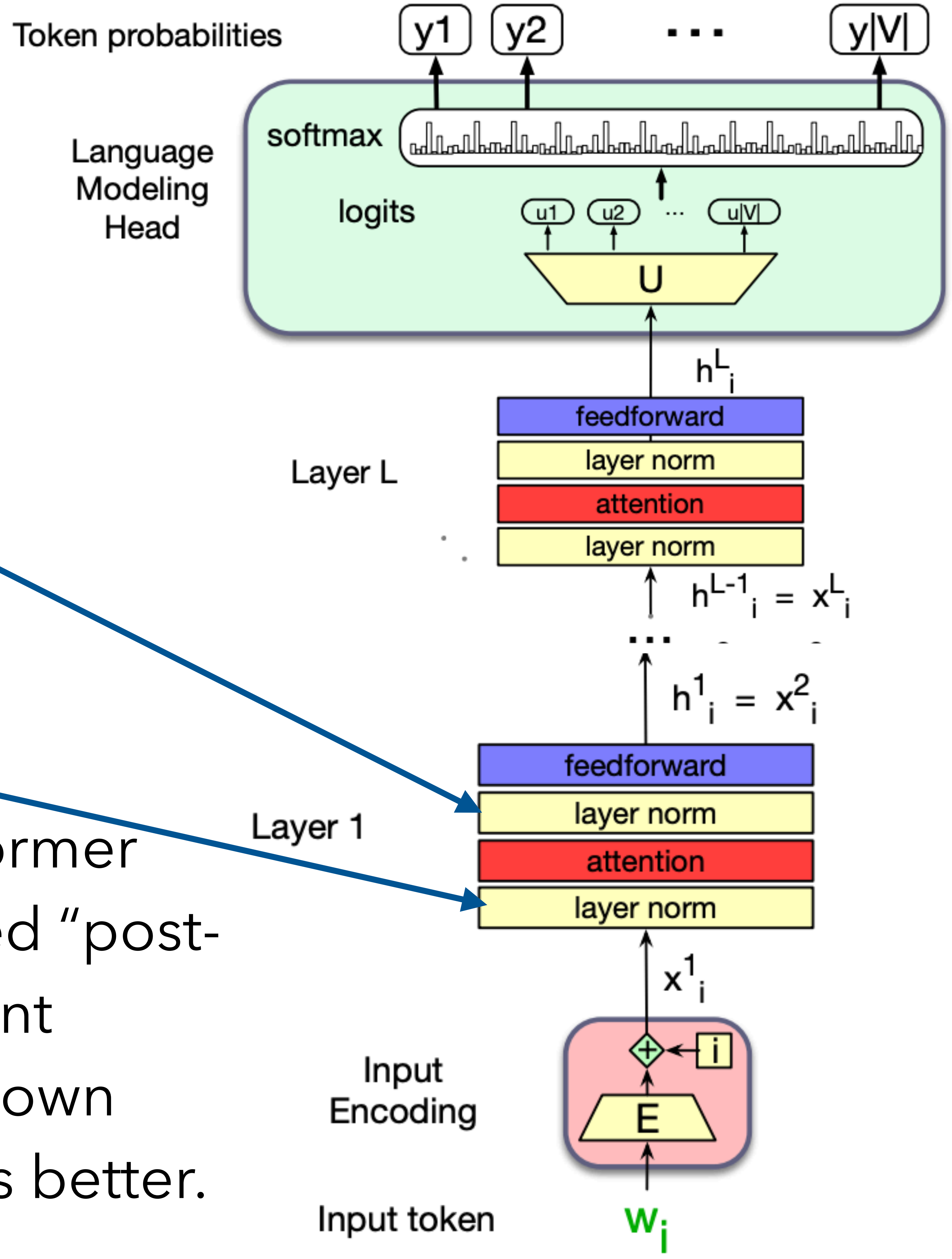
- The actual figure from "Attention is all you need", Vaswani et al, 2017 paper.
- … and the figure you will see everywhere on the internet

Encoder-Decoder Architecture



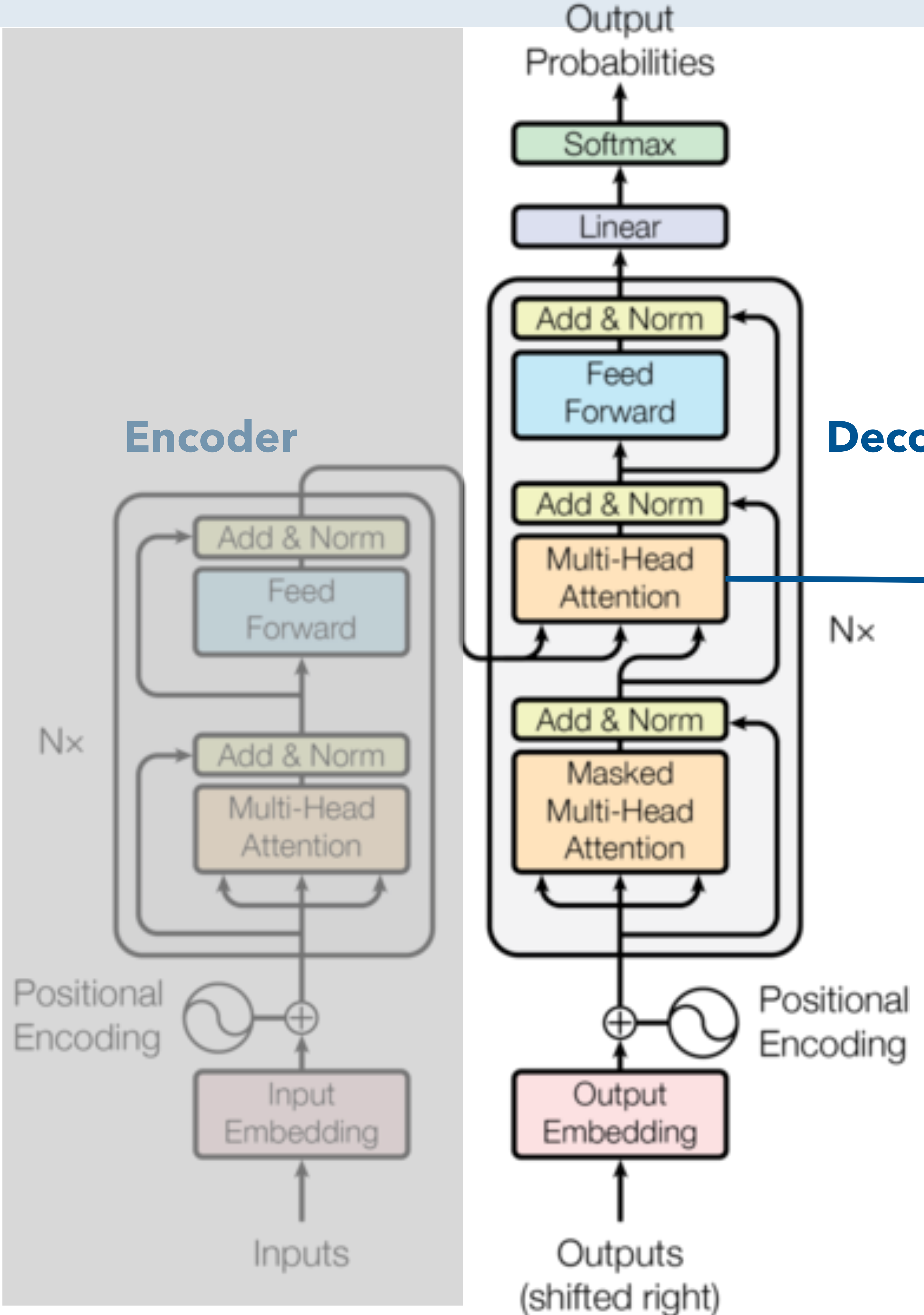
Encoder-Decoder Architecture



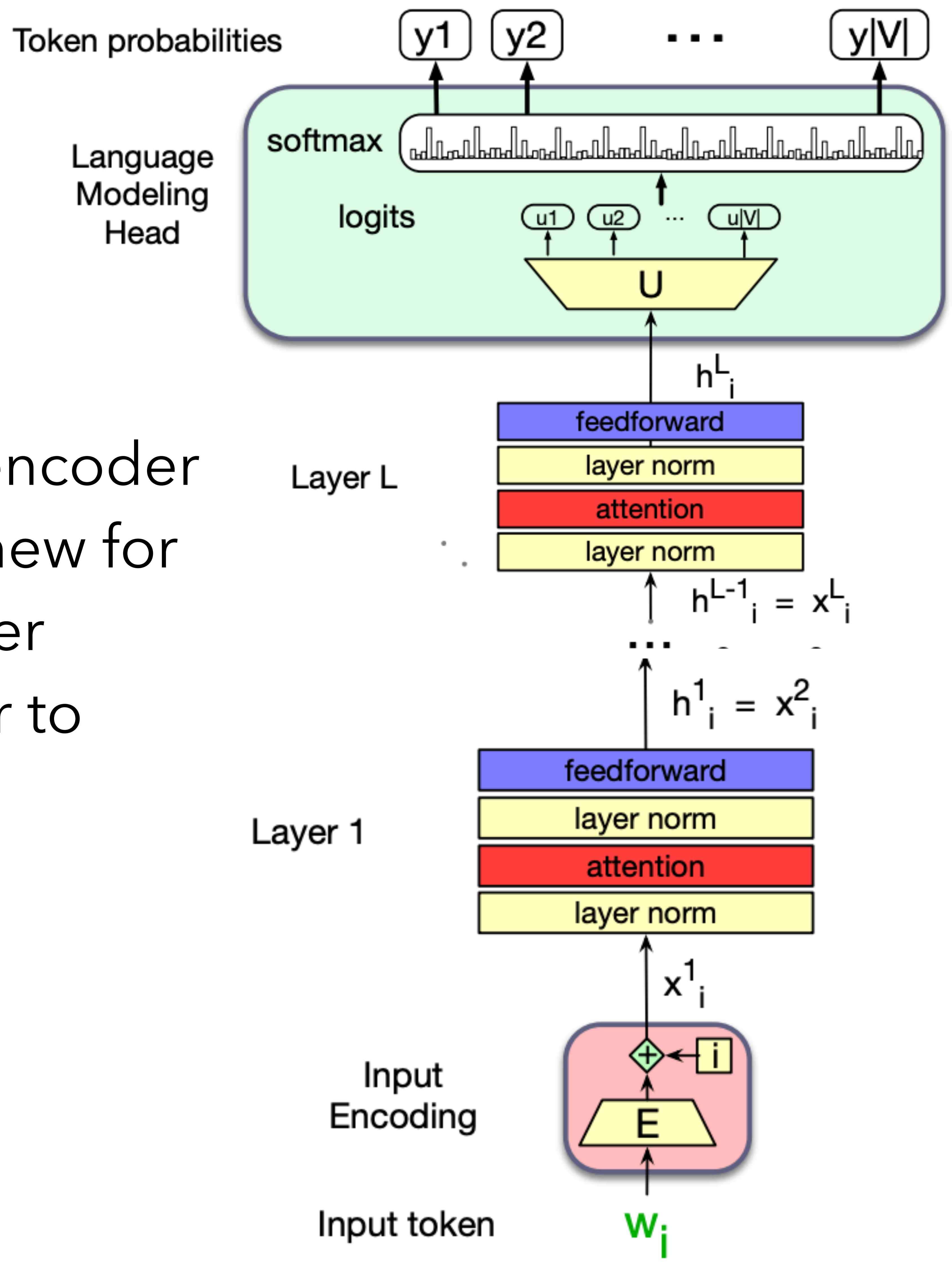


Original transformer model proposed "post-norm", but recent research has shown pre-norm works better.

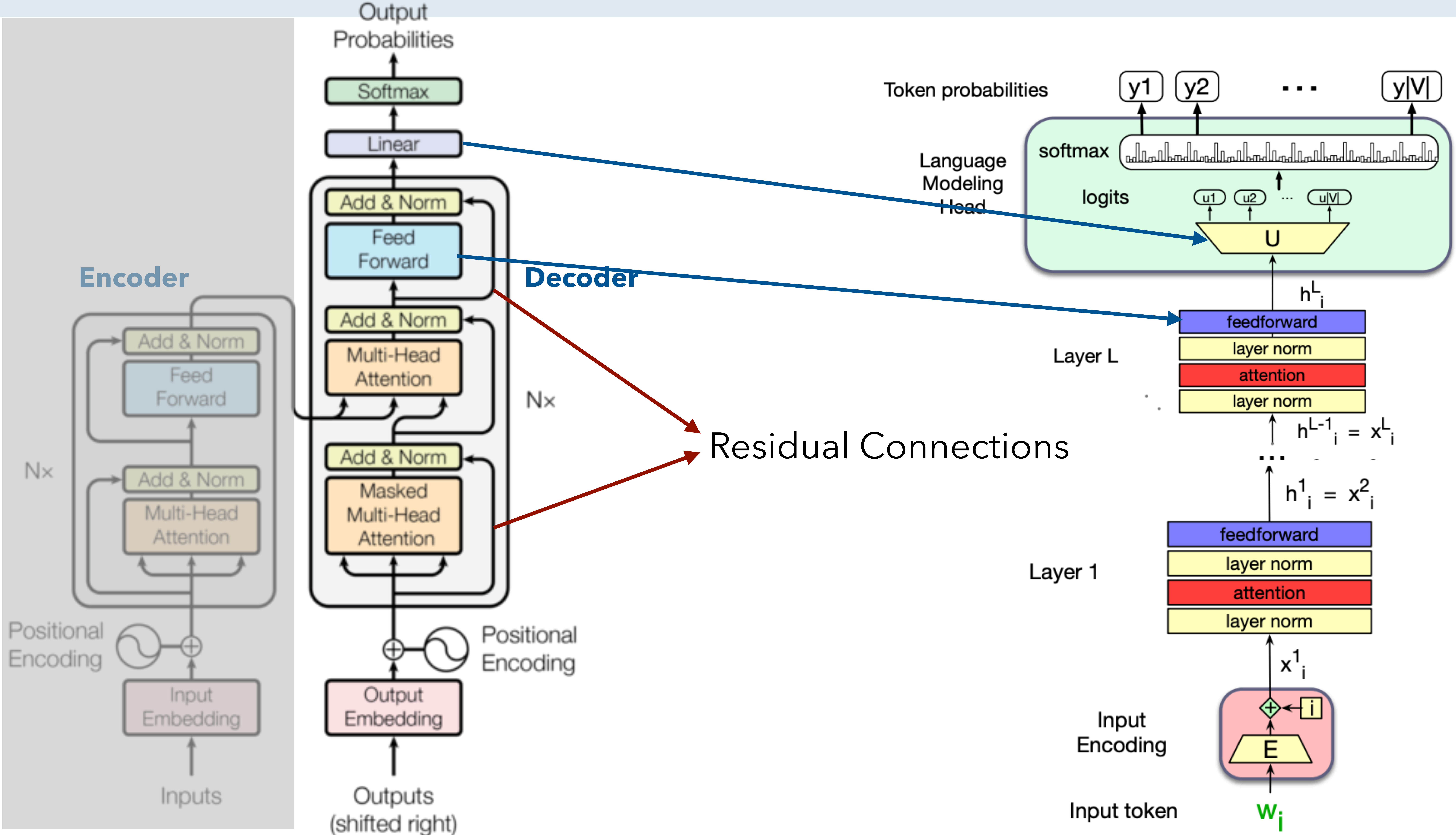
Encoder-Decoder Architecture



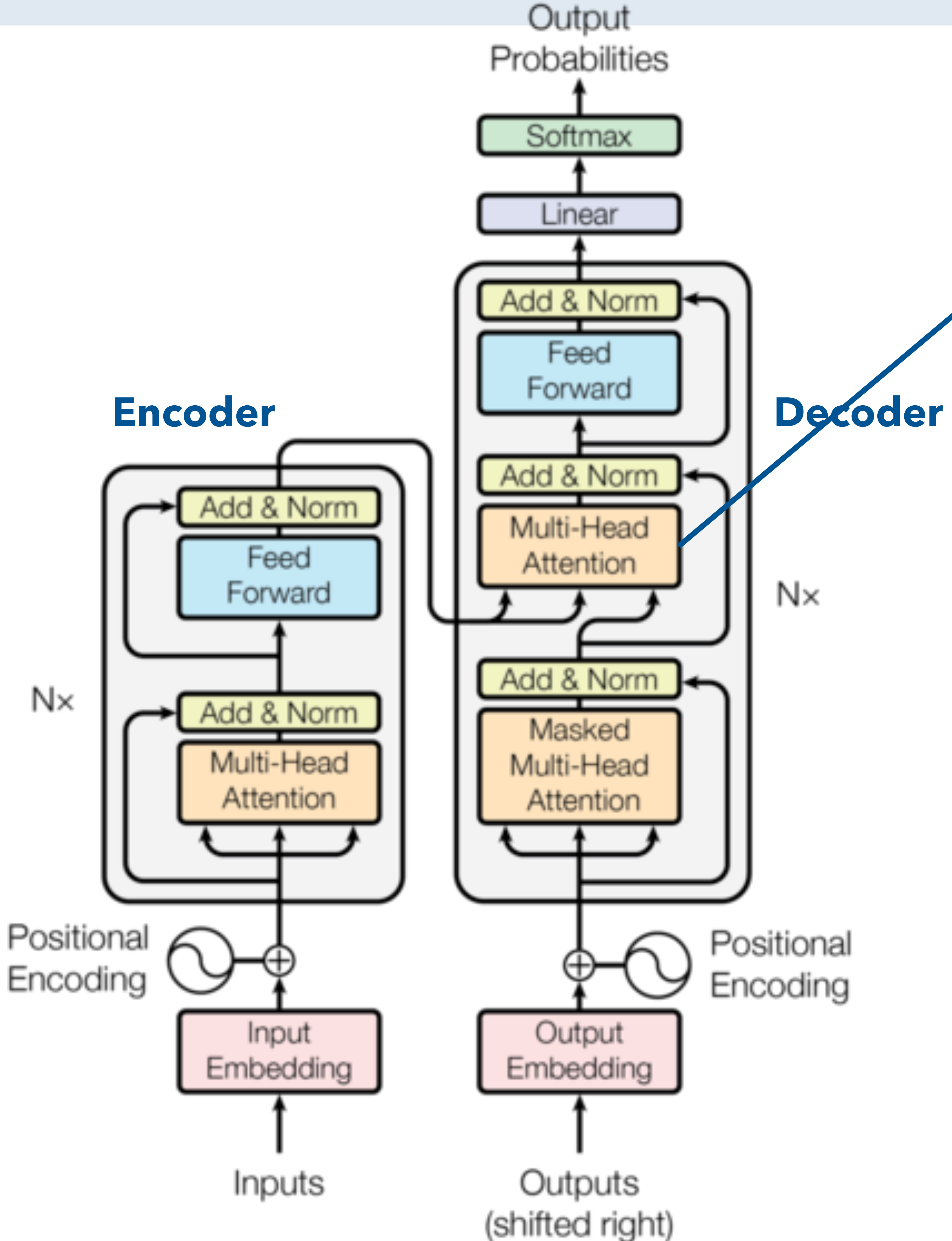
Attention over encoder outputs (this is new for encoder-decoder models!), similar to RNNs



Encoder-Decoder Architecture



Encoder-Decoder Architecture

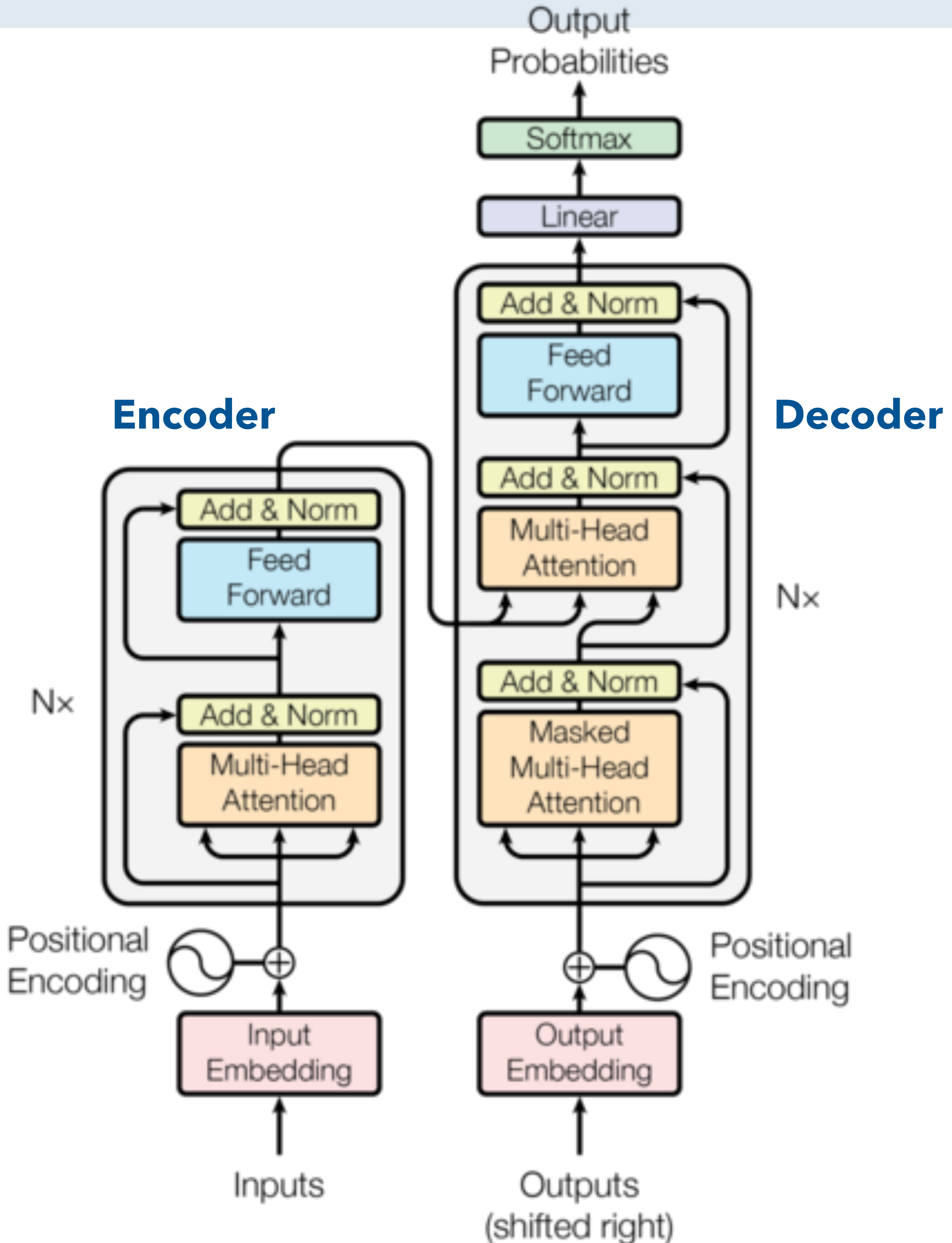


- Multi-head attention from decoder states at each layer to **output of the last layer** of the encoder.

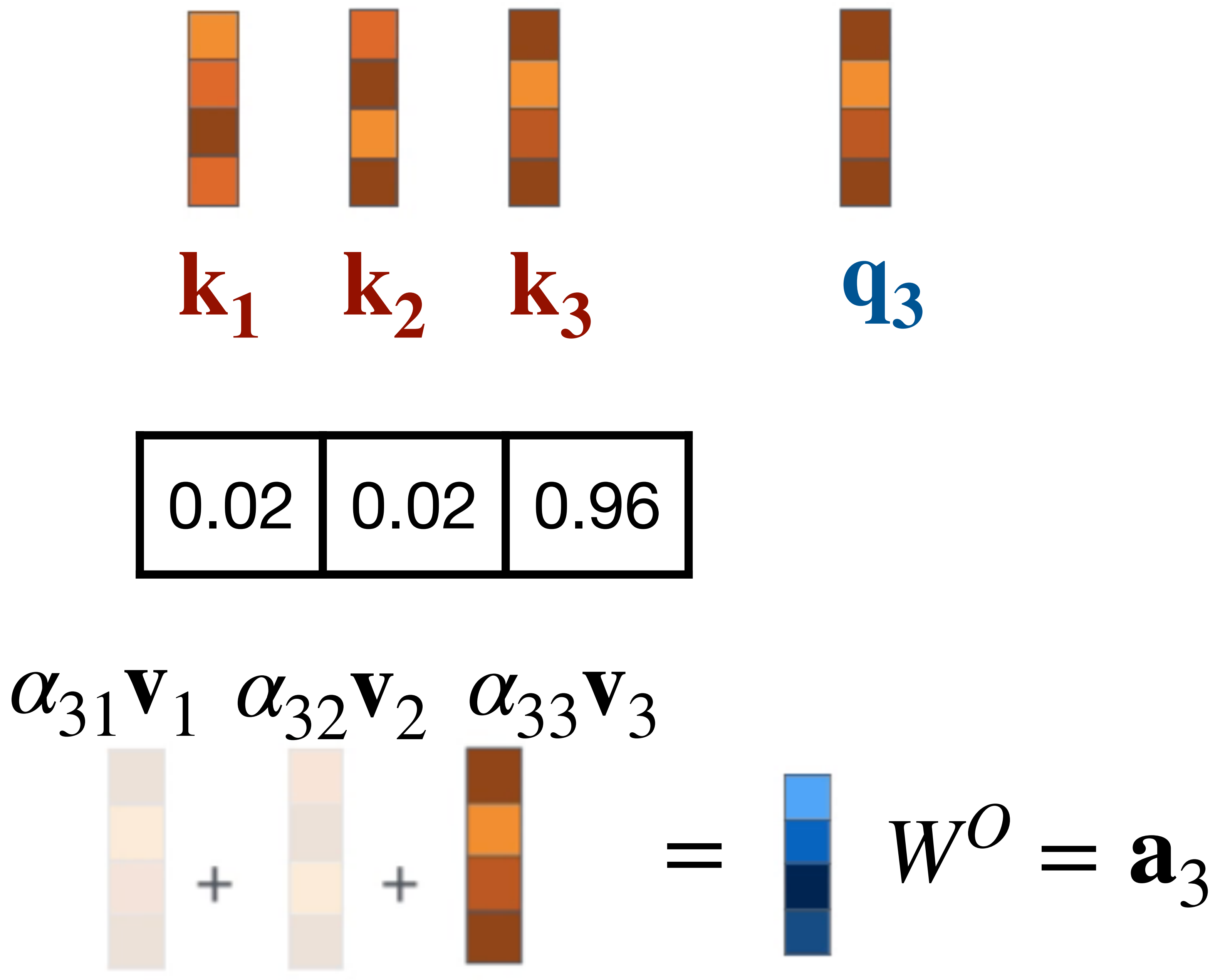
Encoder-Decoder Architecture

Sources: (middle) The Illustrated Transformer
(left and right): Cropped from Attention Is All You Need (AIAYN)

Encoder-Decoder Architecture



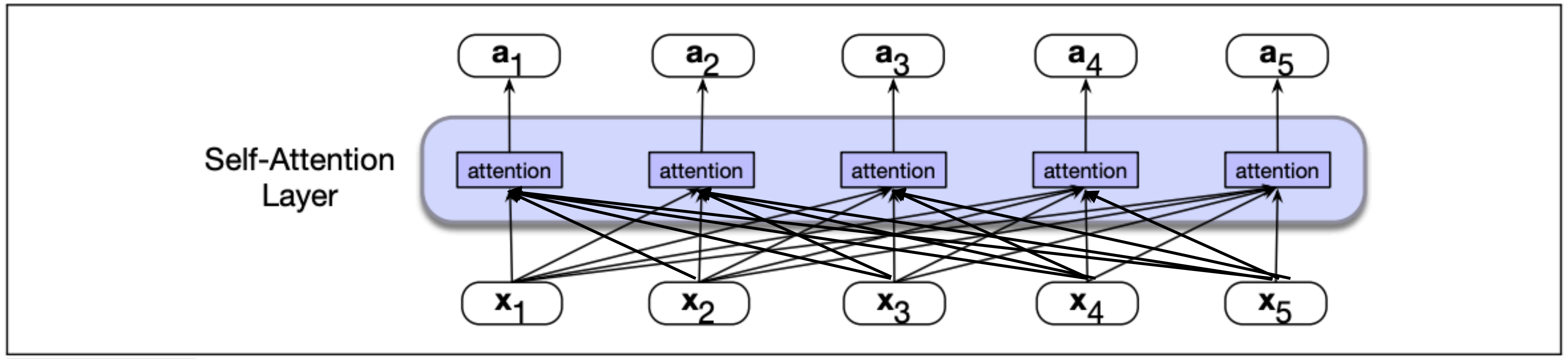
- Recall:

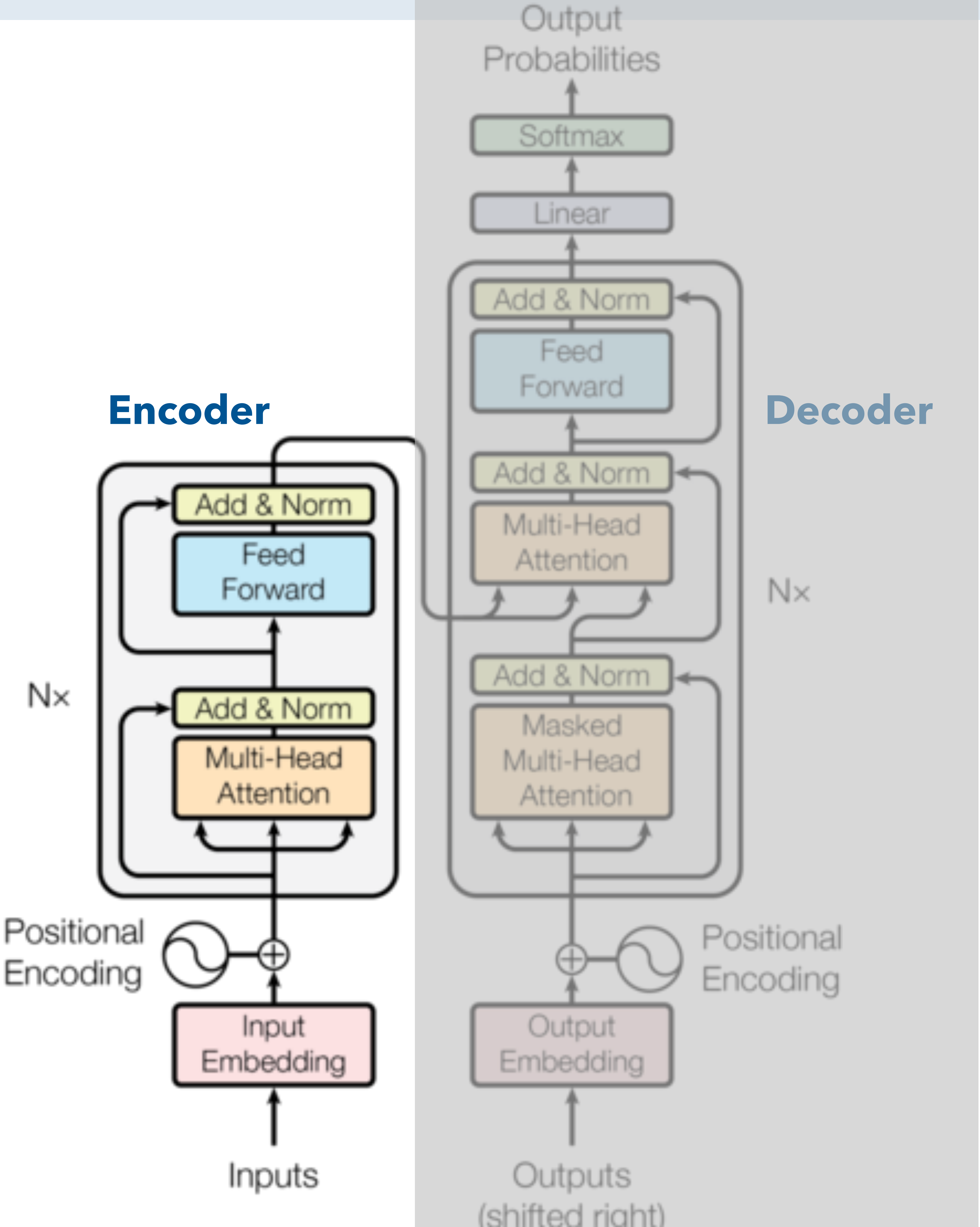


- What is the source of keys, queries, values in attention from decoder to encoder?

Encoder-Decoder Architecture

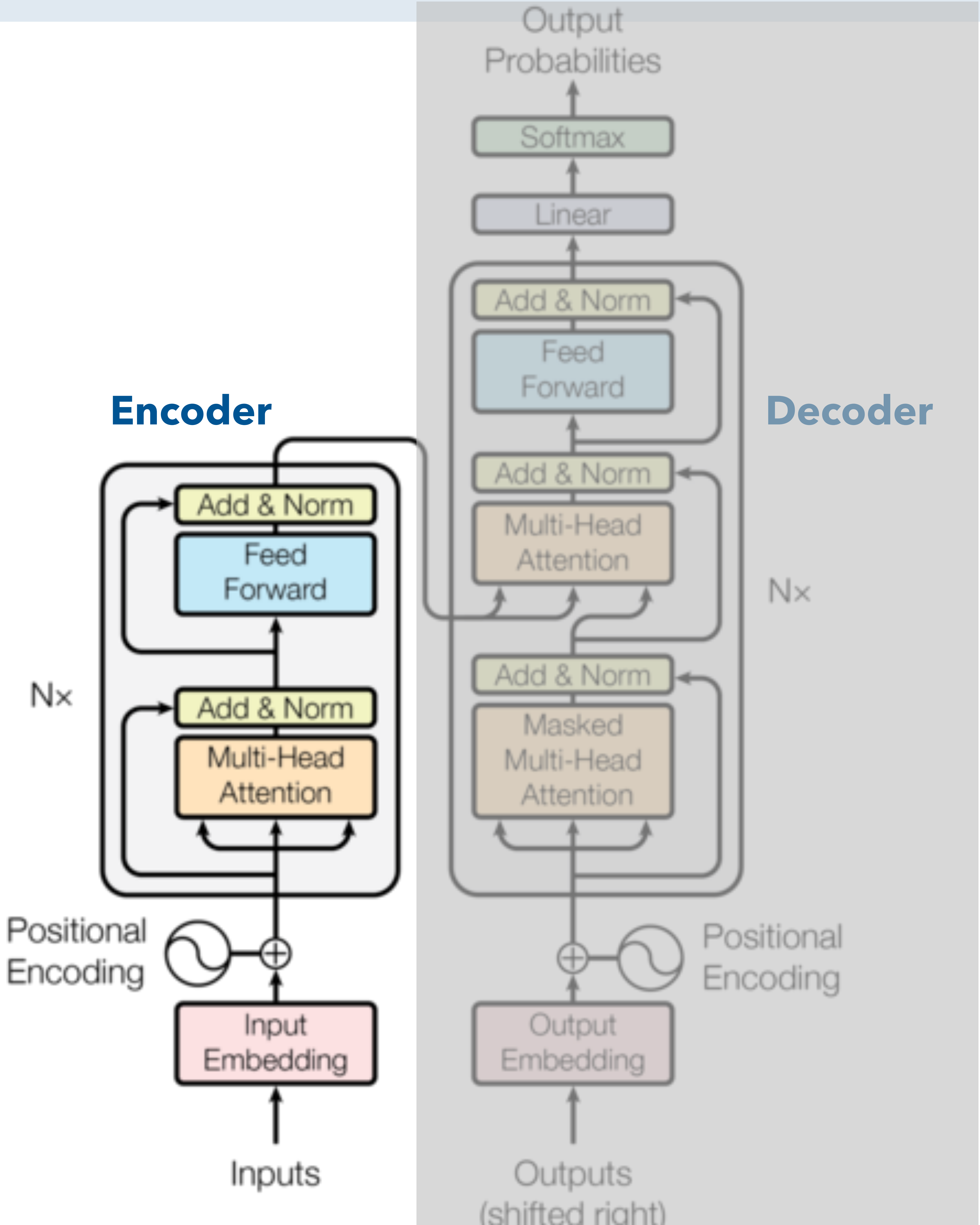
- Encoder architecture is similar to decoder.
- Only difference: This attention is **not** causal.
    - All tokens attend to all other tokens.





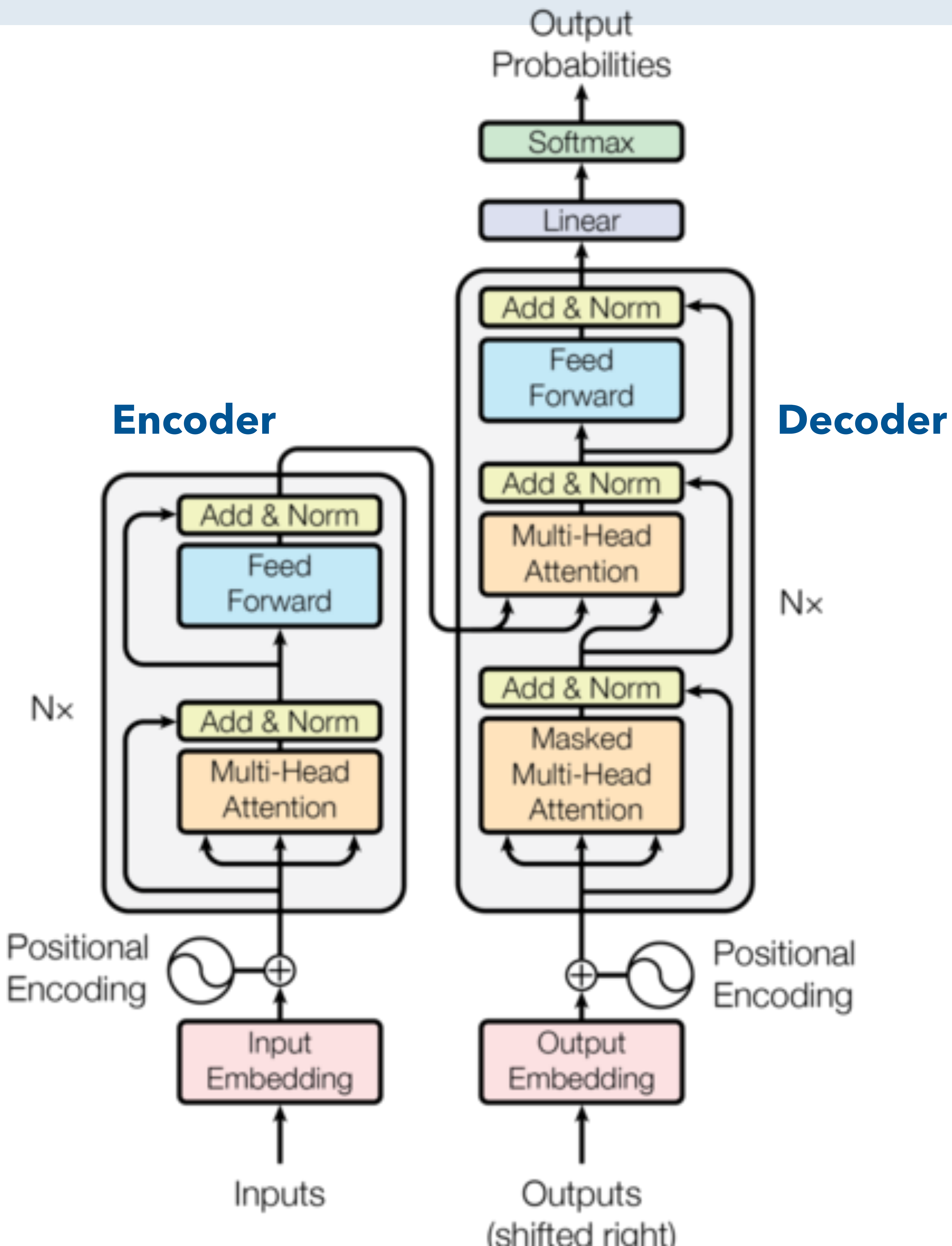
Encoder-Decoder Architecture

- Encoder architecture is similar to decoder.
- Only difference: This attention is **not** causal.
  - All tokens attend to all other tokens.
- At each time step, the encoder output h_t can be viewed as a "contextual" representation of the input word w_t.



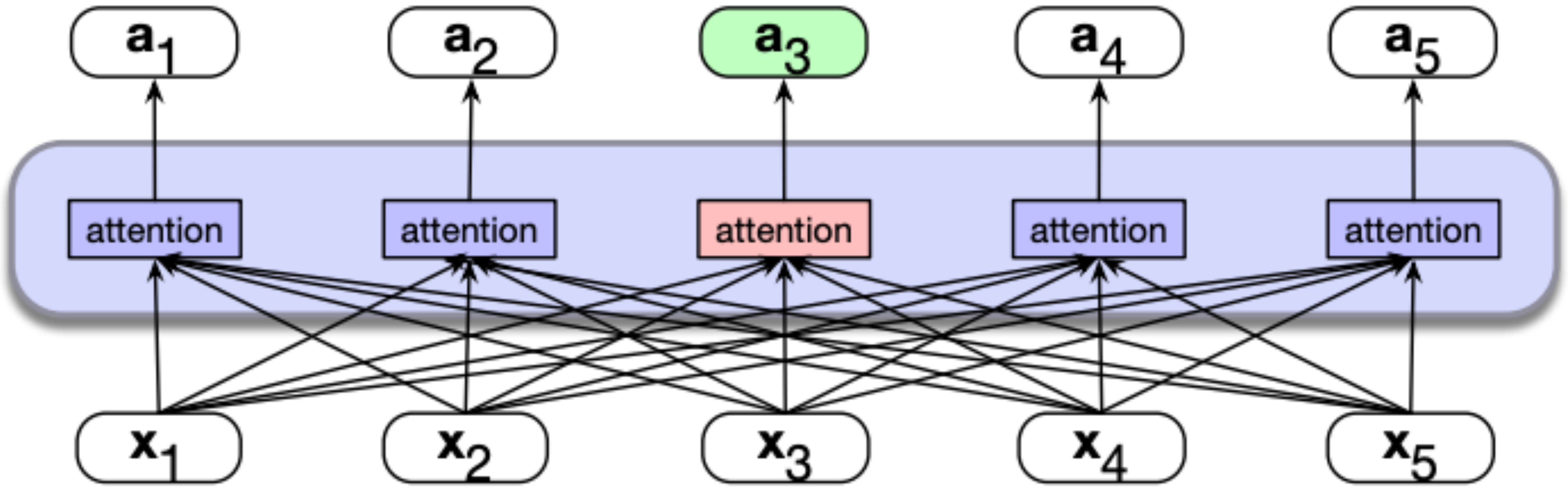
Encoder-Decoder Architecture

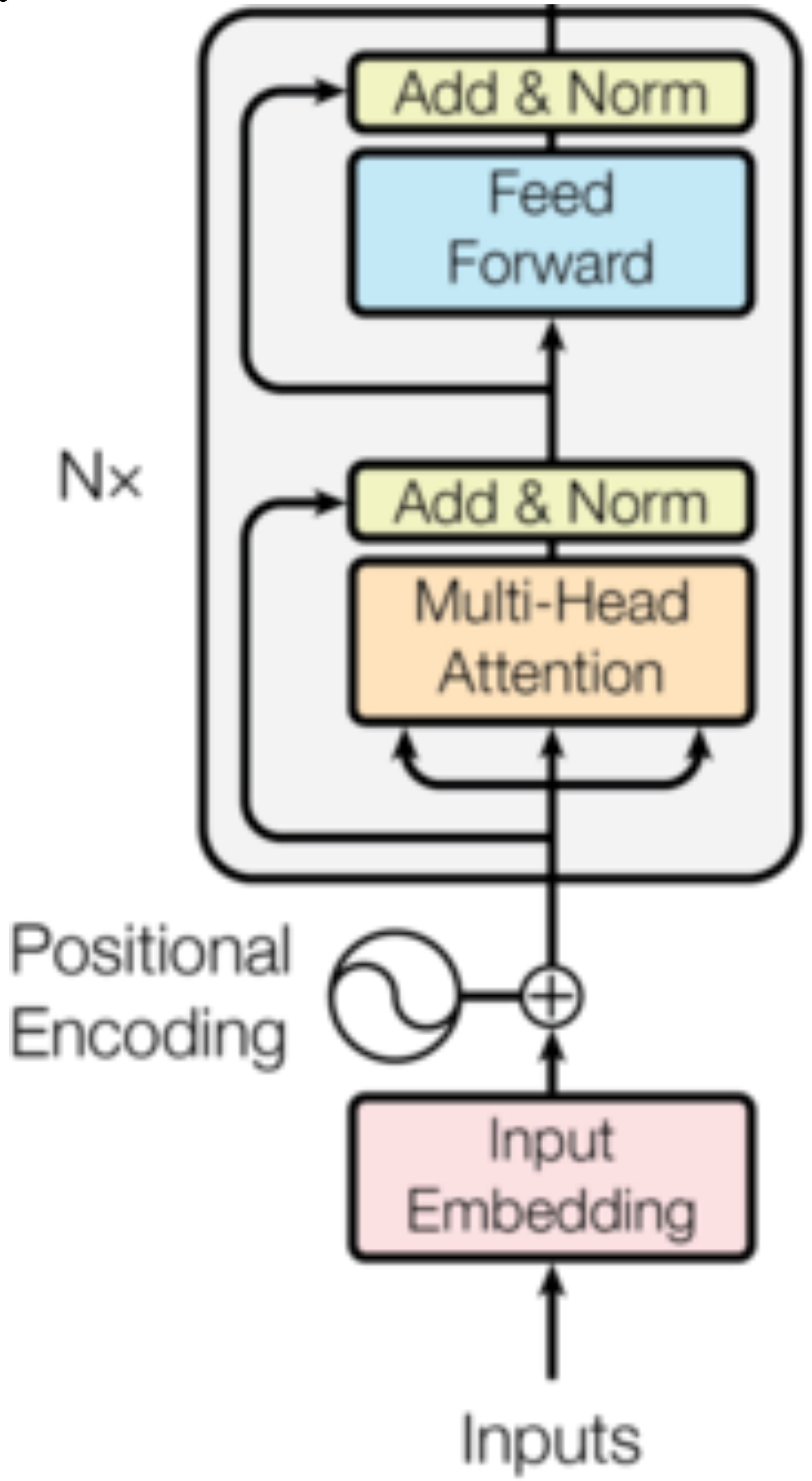
- Training
  - Next-token prediction at the decoder output.



Encoder-only architectures

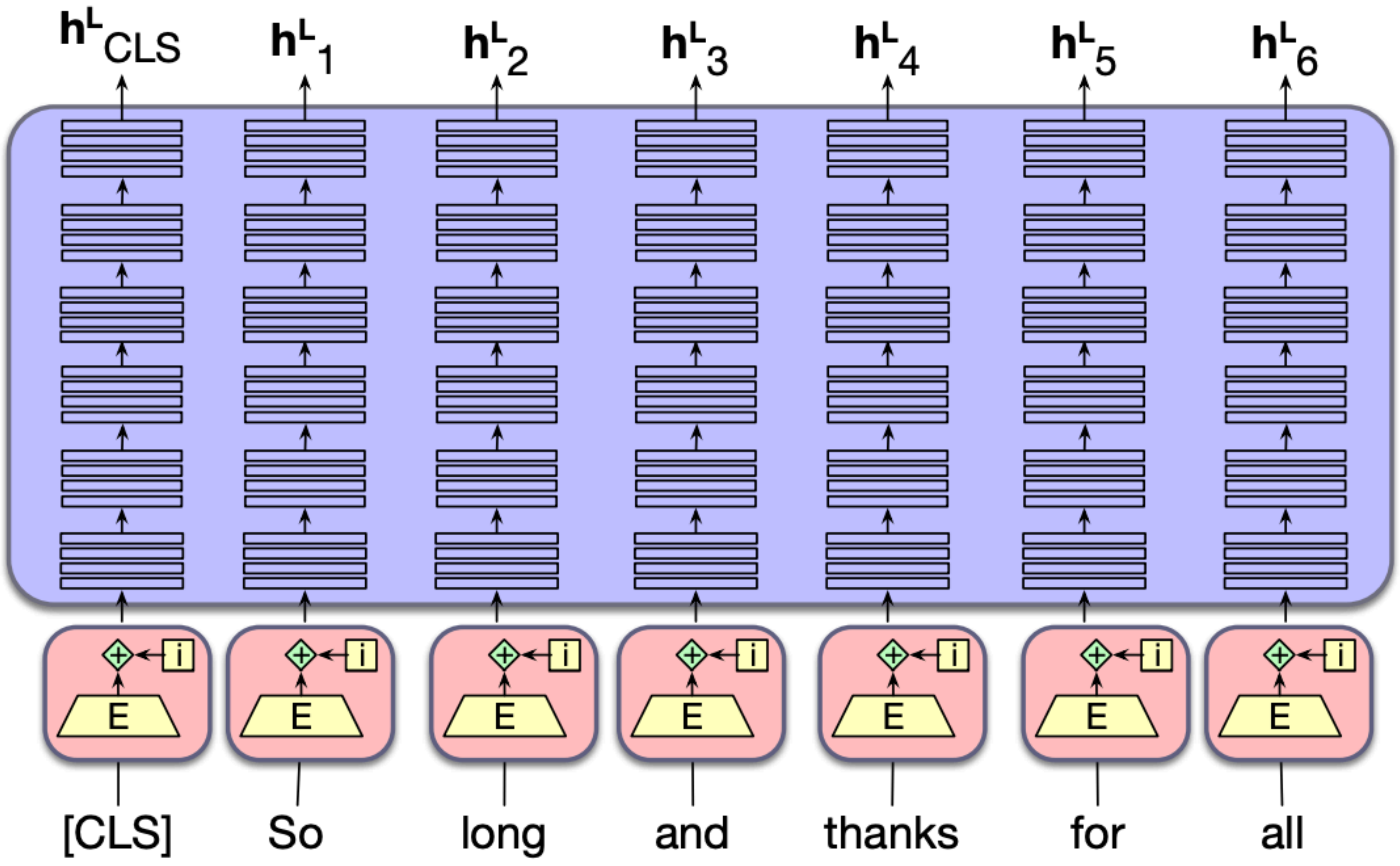
- Same architecture components as the decoder.
- **Difference:** the attention is bidirectional, not causal





Encoder-only Architecture

- What is this useful for?
  - Gives contextual representations for each input word.

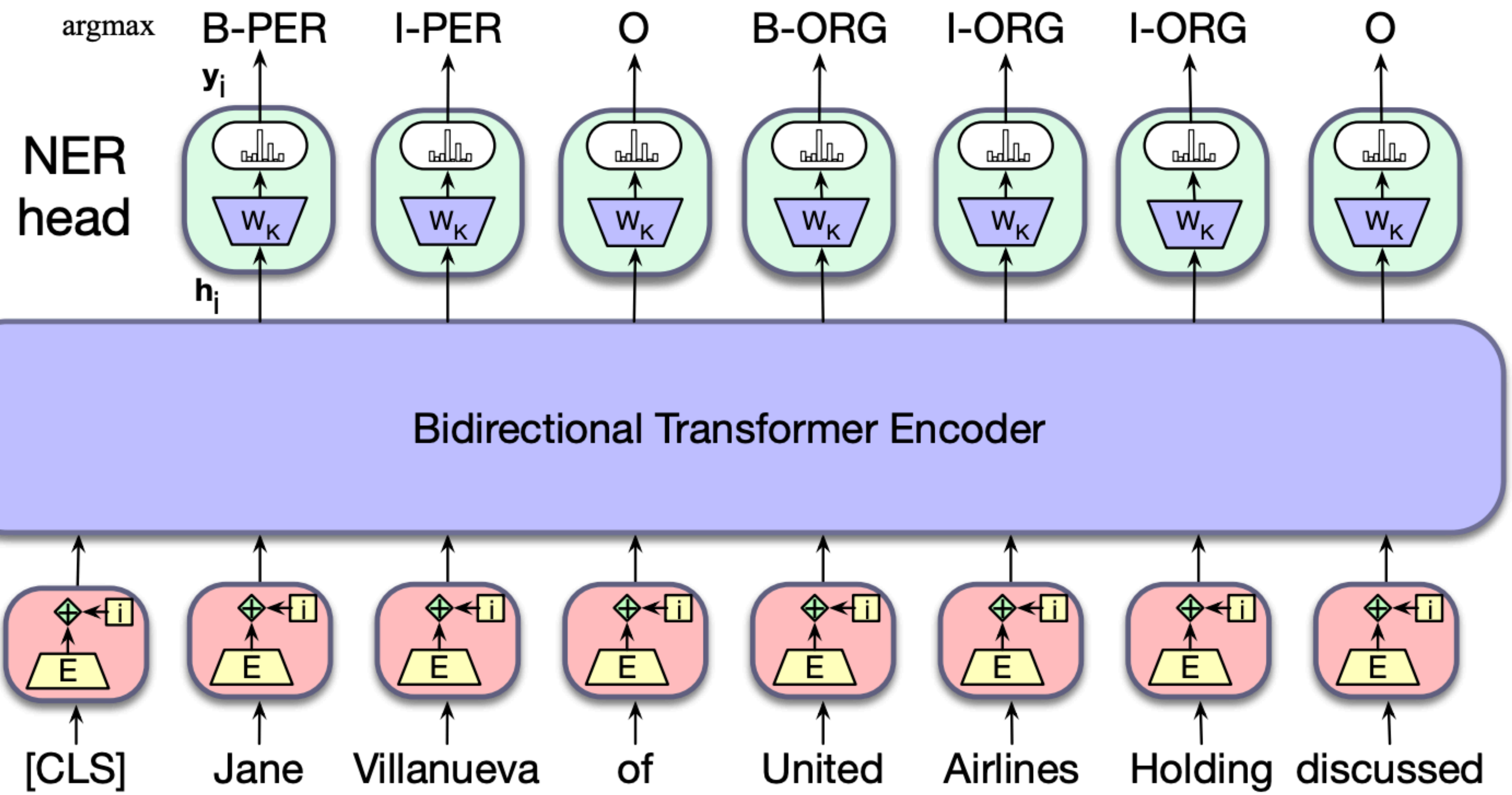


Encoder-only Architecture

- Word sense disambiguation
  - A sense (or word sense) is a discrete representation of one aspect of the meaning of a word

Encoder-only Architecture

- Classification?
  - Sequence Labeling

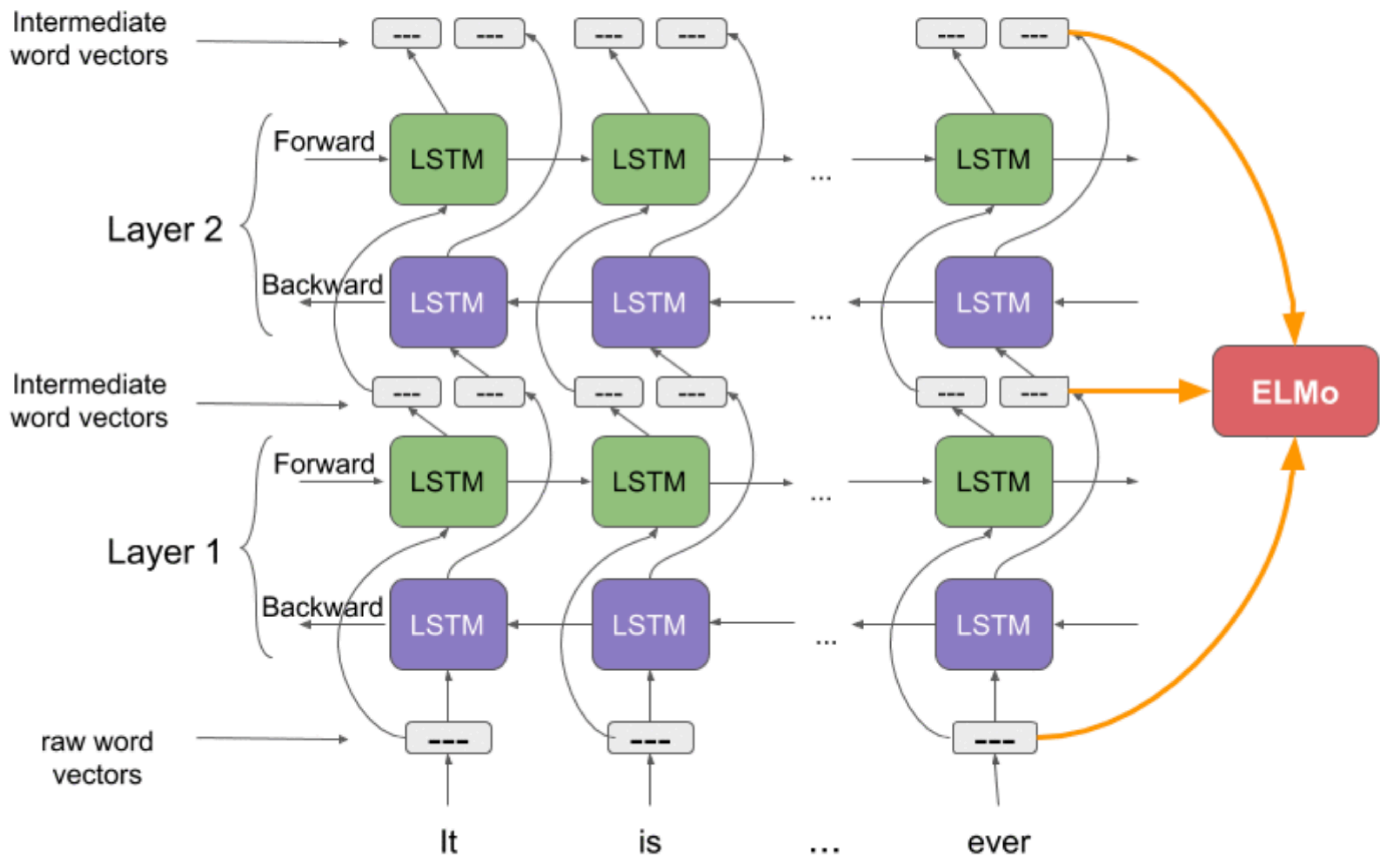


Why are these more powerful than bidirectional RNNs?

- Predecessor: ELMo (Peters et al., 2018)

Uses bi-LSTM-based encoder-decoder.

Combines hidden vectors from different layers.



Why are these more powerful than bidirectional RNNs?

- BiLSTMs are not quite the same as full self-attention:

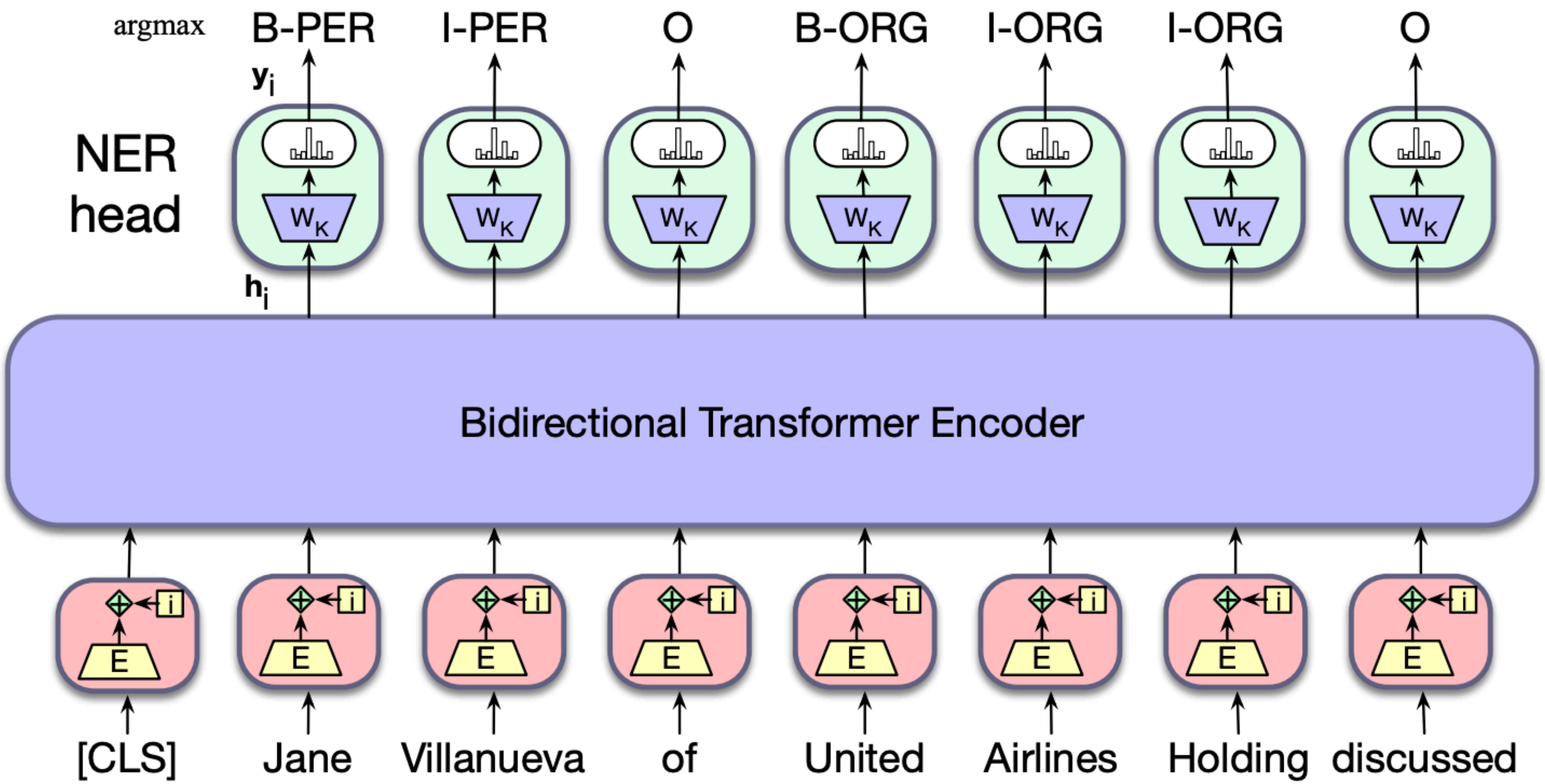
Try to predict the italicized word from just the left or just the right context.
The celebrity , Michael ***Jordan*** , was a player in the NBA .

A left-to-right model could also answer “Jackson” and be sensible but wrong. (Reference to singer/songwriter Michael Jackson)

A right-to-left model could also answer “Curry” and be sensible but wrong. (Reference to NBA player Steph Curry)

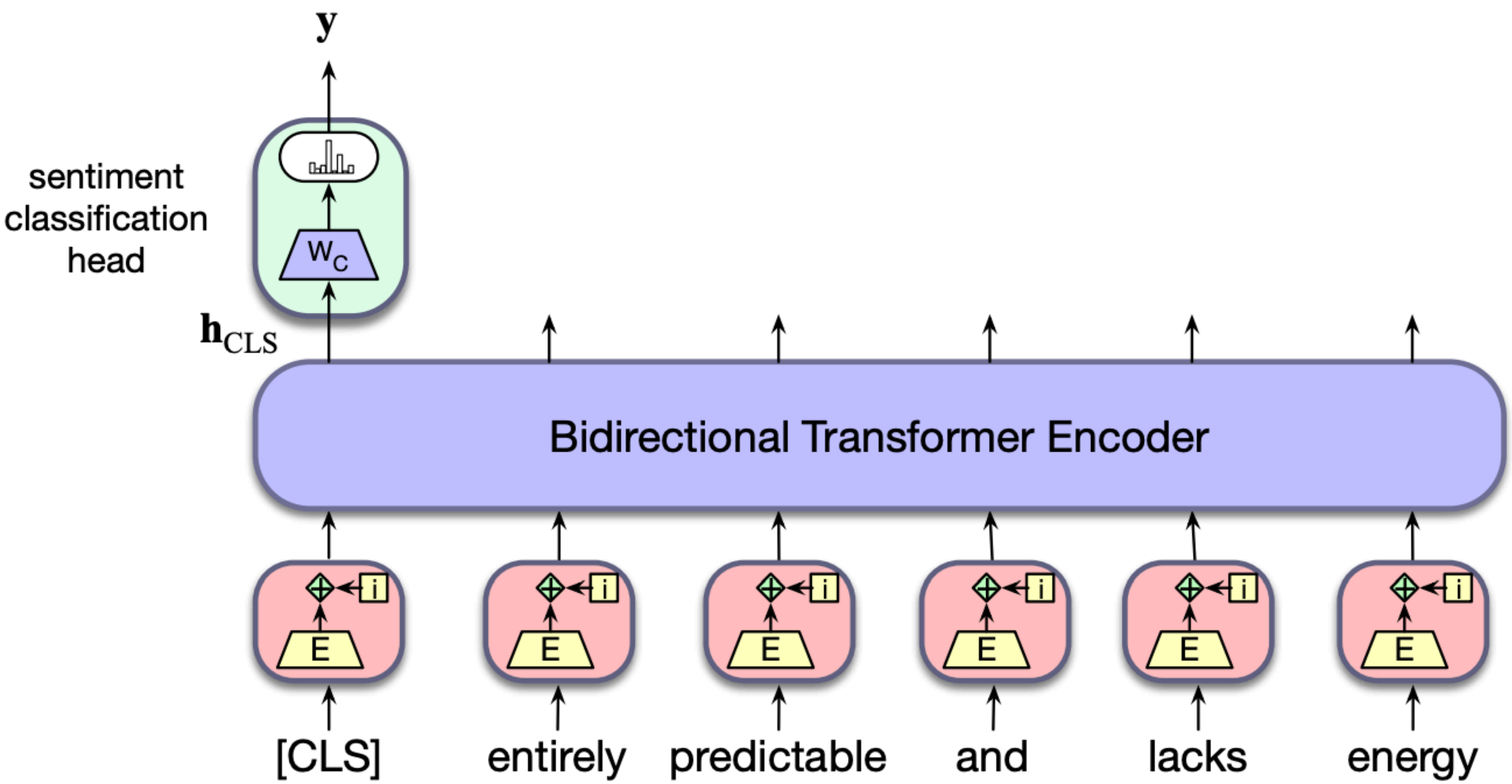
Encoder-only Architecture

- Classification?
  - Sequence Tagging



Encoder-only Architecture

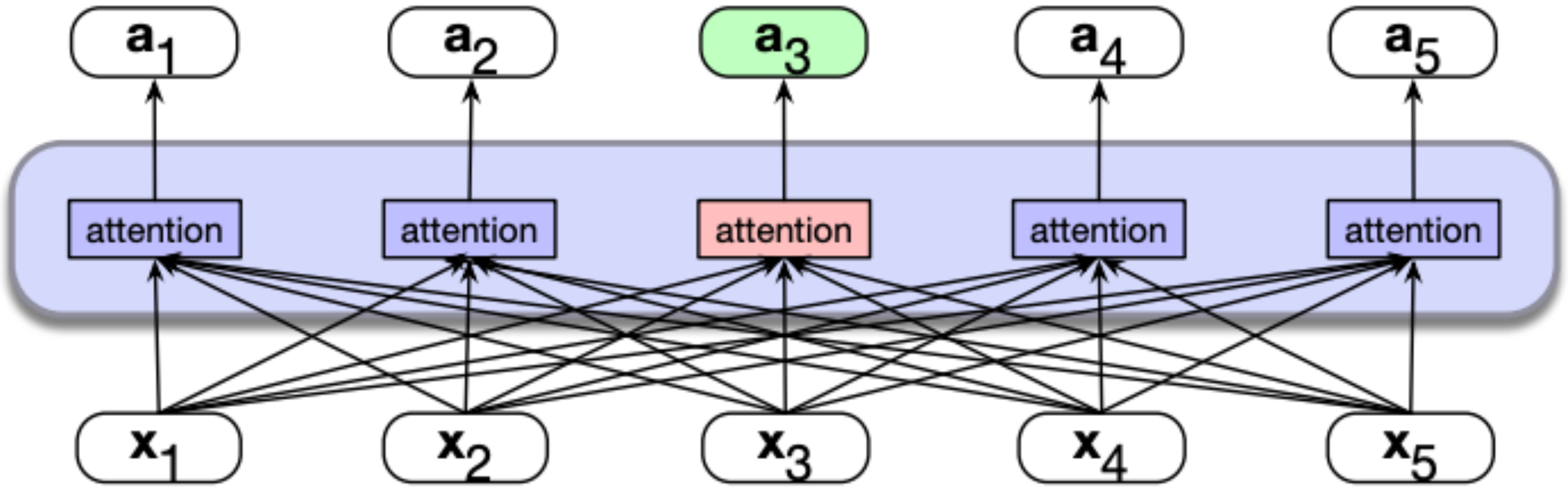
- Classification?
  - Sequence-level classification

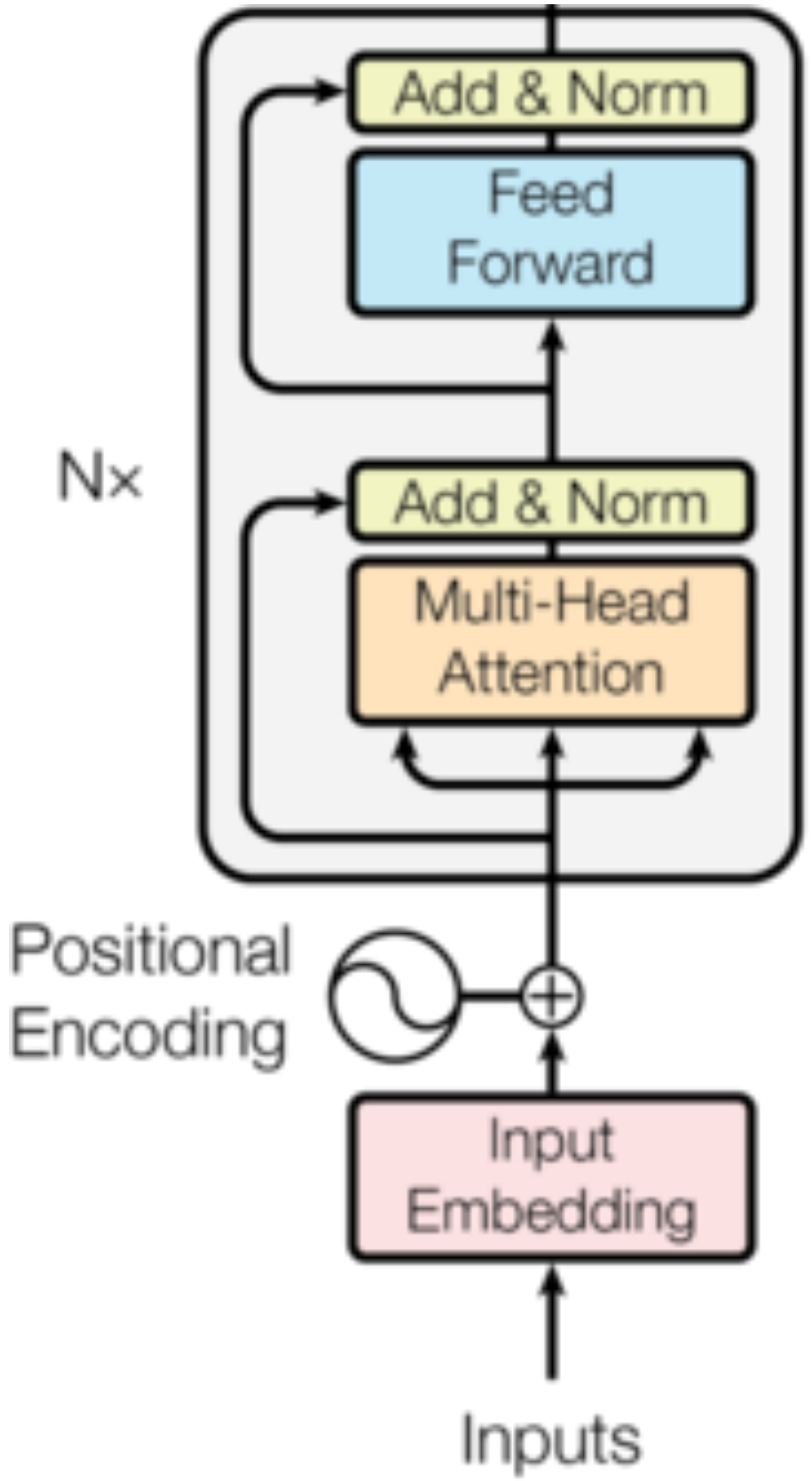


Encoder-only architectures

- Can we use the same next-token prediction task to train encoder models?

No! The desired output is part of the input!





Slide Acknowledgements

- Earlier versions of this course offerings including materials from Claire Cardie, Marten van Schijndel, Lillian Lee.