Encoder-Decoder (seq2seq) Models

- Can view many tasks as mapping from an input sequence of tokens to an output sequence of tokens

- Syntactic parsing

The dog ran ⟶ `(S (NP (DT the) (NN dog) ) (VP (VBD ran) ) )`

- Semantic parsing:

What states border Texas ⟶ `λ x state( x ) ∧ borders( x , e89 )`

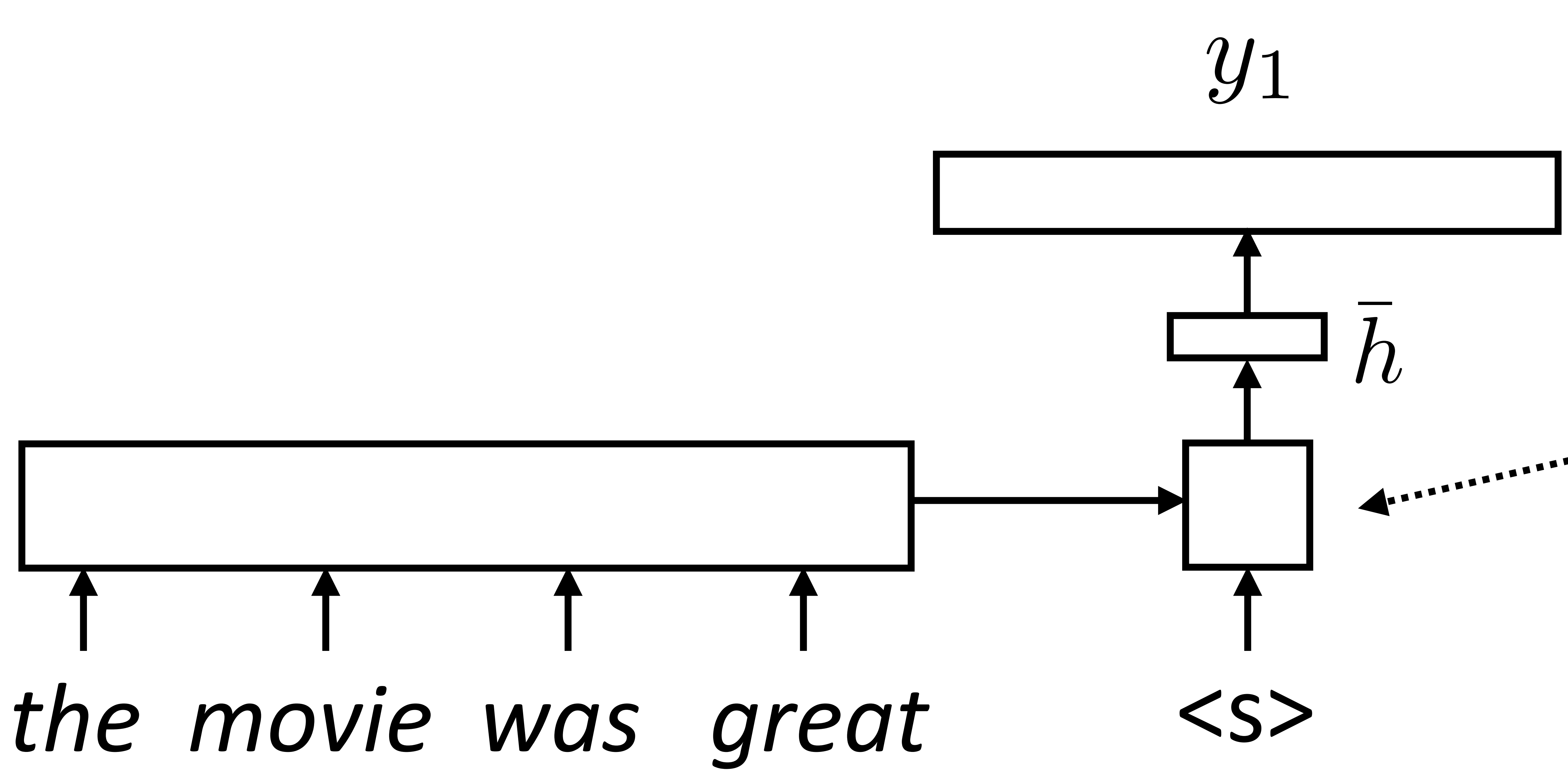
- Machine translation, summarization, dialogue can all be viewed in this framework as well; our examples will be MT for now

- This is slightly different from language modeling ("decoder-only") because the input and output vocabularies can be different. Modern language models like ChatGPT can model all this with a shared vocabulary.

Seq2seq Models

- Generate next word conditioned on previous output as well as input

- W size is |vocab| x |hidden state|, softmax over entire vocabulary



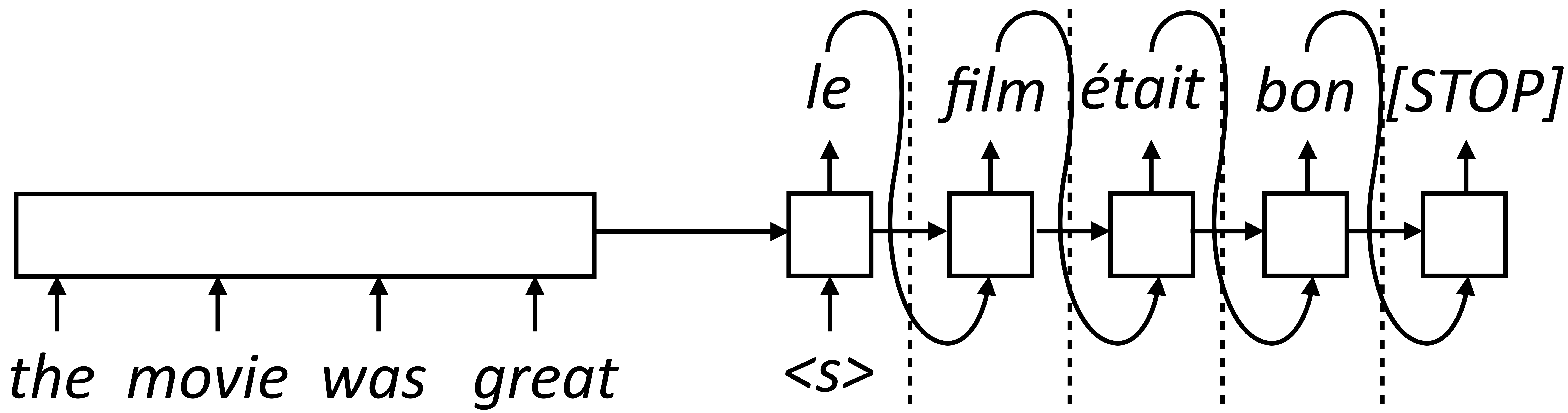
$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W\bar{h})$$

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1})$$

Decoder has separate parameters from encoder

- Example: translate this input **x** into a French output **y**

Seq2seq Models: Inference and Training

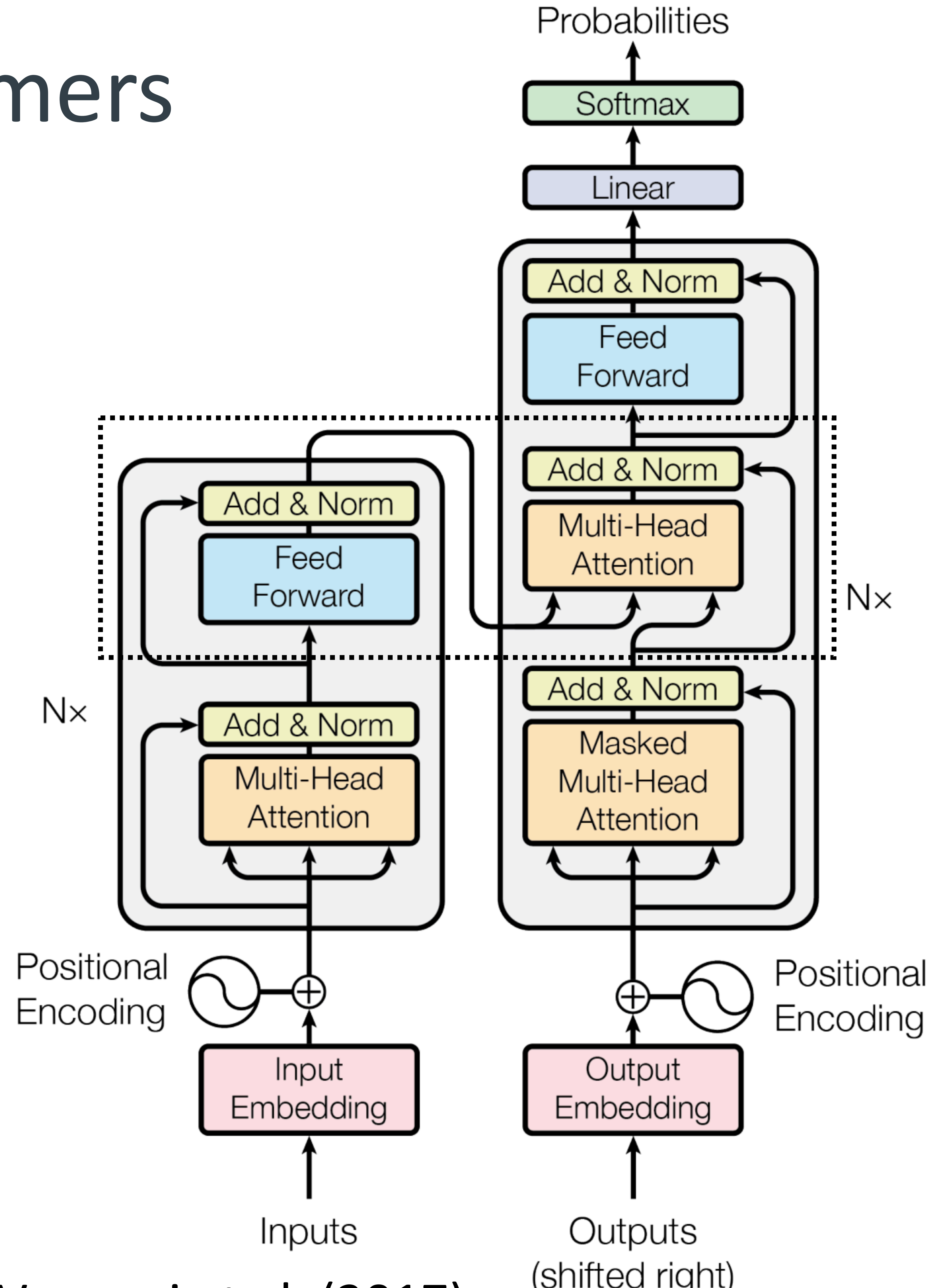


- **Inference**: need to compute the argmax over the word predictions and then feed that to the next Transformer call
- Decoder is advanced one state at a time until [STOP] is reached
- The encoder can just be run a single time

- **Training**: same as language model training, maximize the probability of the gold sequence ***y*** (now conditioned on the input ***x***)

Seq2seq Transformers

- Encoder-decoder Transformer includes a separate multi-head attention computation that attends to the encoder inputs
- Otherwise, behaves very similarly to the Transformer we've seen before



Vaswani et al. (2017)