NLP with MxNet: Bring your own Container

Wen-ming Ye
Solutions Architect
AWS

Rachel Hu
Applied Scientist
AWS

Laurens ten Cate
ProServe Data Scientist
AWS

AWS re:Invent

Agenda

Logistics (10 min)

Natural Language Processing Background (20 min)

State of the Art: BERT (20 min)

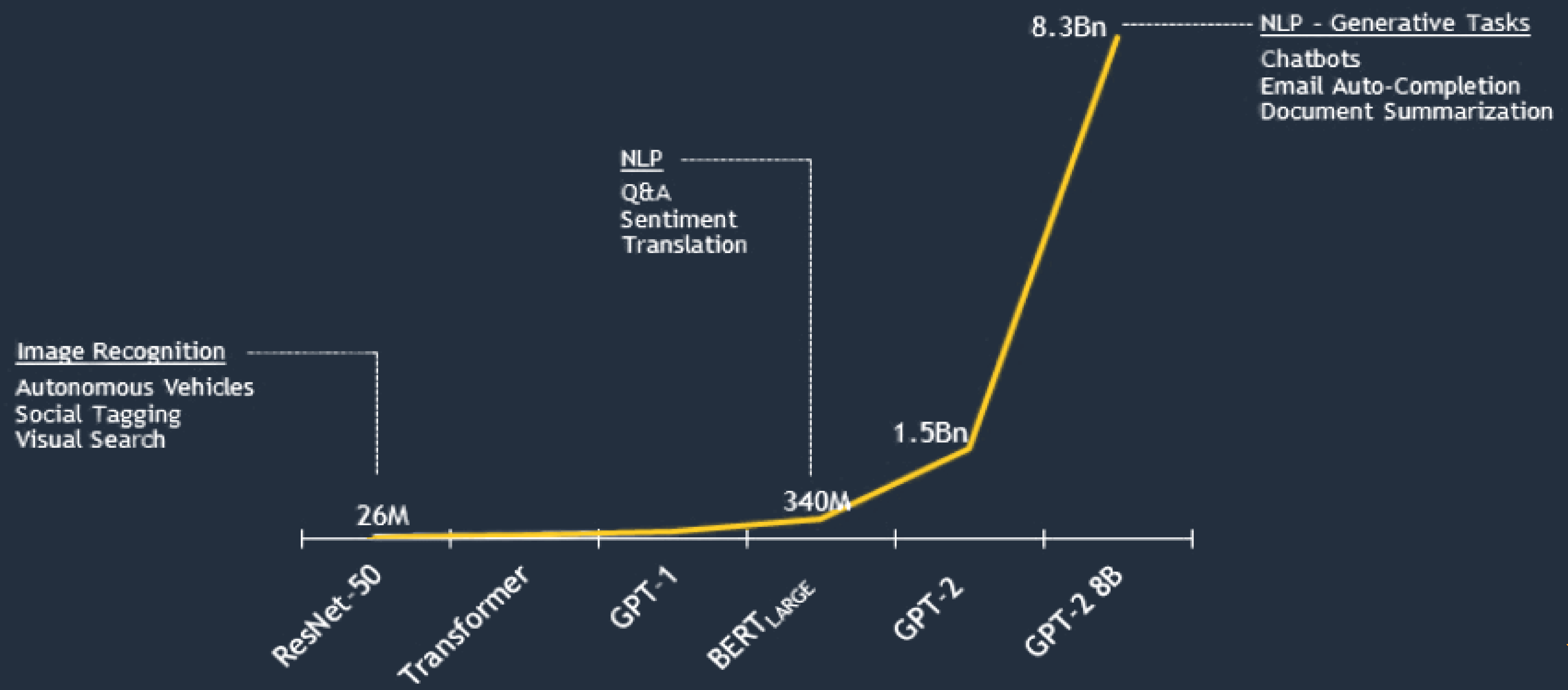
Lab 1: BERT Sentiment model + Q&A model (30 min) (Level 300)

Lab 2: End-to-end Deployment on SageMaker (1 hour) (Level 400)

Learning Resources

Exploding model complexity

Number of parameters by network



Current State of Natural Language Processing

Text synthesis

Content: Two dogs play by a tree.

Style: happily, love

Synthesized text: Two dogs **in love** play **happily** by a tree.

Natural language processing nowadays

Question answering

Question: Who shall use GluonNLP?

Passage context: GluonNLP provides implementations of the state-of-the-art (SOTA) deep learning models in NLP, and build blocks for text data pipelines and models. It is designed for engineers, researchers, and students to fast prototype research ideas and products based on these models.

Tokenization – basic strategies

“Buy 5000m3 Dec CFT, WTI -9.25, Equinor US, OX, USD, B McBride”

Character level

['B', 'u', 'y', ' ', '5', '0', '0', '0', 'm', '3', ' ', 'D', 'e', 'c', ' ', 'C', 'F', 'T', ',', ' ', 'W', 'T', 'I', ' ', '-', '9', '.', '2', '5', ',', ' ', 'E', 'q', 'u', 'i', 'n', 'o', 'r', ' ', 'U', 'S', ',', ' ', 'O', 'X', ',', ' ', 'U', 'S', 'D', ',', ' ', 'B', ' ', 'M', 'c', 'B', 'r', 'i', 'd', 'e']

Word Level

['Buy', '5000m3', 'Dec', 'CFT,', 'WTI', '-9.25,', 'Equinor', 'US,', 'OX,', 'USD,', 'B', 'McBride']

Advanced Tokenization

Byte-pair encoding (recursively) reduces byte-pairs to new bytes to both compress and reduce commonly occurring byte-pairs.

"environment, mentally" → [65 6e 76 69 72 6f 6e 6d 65 6e 74 2c 20 6d 65 6e 74 61 6c 6c 79]

So 'ment' or [6d 65 6e 74] would become a new byte.

- Reduces out-of-vocab issues with new words due to possible fallback on single bytes
- Captures shared semantic information of sub-word token

Result (in utf-8 encoded bytes)

['▁buy', '▁0000', 'm', '0', '▁oct', '▁c', 'pr', ',', '▁w', 'ti', '-0,', '▁tour', 'mal', 'ine', '▁oil', ',', '▁spect', 'ron', '▁uk', ',', '▁us', 'd', ',', '▁b', '▁cole']

Grapheme/token representation in NLP

- Define words as a vector
  - bag-of-words approach where sentence is sum of word vectors.

$$
\begin{array}{c}
\begin{array}{cccc} \text{Deep} & \text{learning} & \text{is} & \text{fun} \end{array} \\
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{array}
$$

Limitations: no semantic information

remember:

$$\text{-}L2\ Norm = ||v||_2 = \sqrt{\sum_{k=1}^{n} x_k{}^2}$$

$$\left\| \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\end{bmatrix} - \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\end{bmatrix} \right\|_2 = \left\| \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\end{bmatrix} - \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\end{bmatrix} \right\|_2 = \sqrt{2}$$

$$|| v_{automobile} - v_{car} ||_2 = || v_{automobile} - v_{mountain} ||_2 = \sqrt{2}$$

Ideally we would want:

$$|| v_{automobile} - v_{car} ||_2 \approx 0$$

Embedding

- Word embeddings
  - Vector representations of words
- Word2Vec (shallow word embeddings)
  - Training
    - Models central words given context words

Deep **learning** is fun!

P(learning | deep, is, fun)

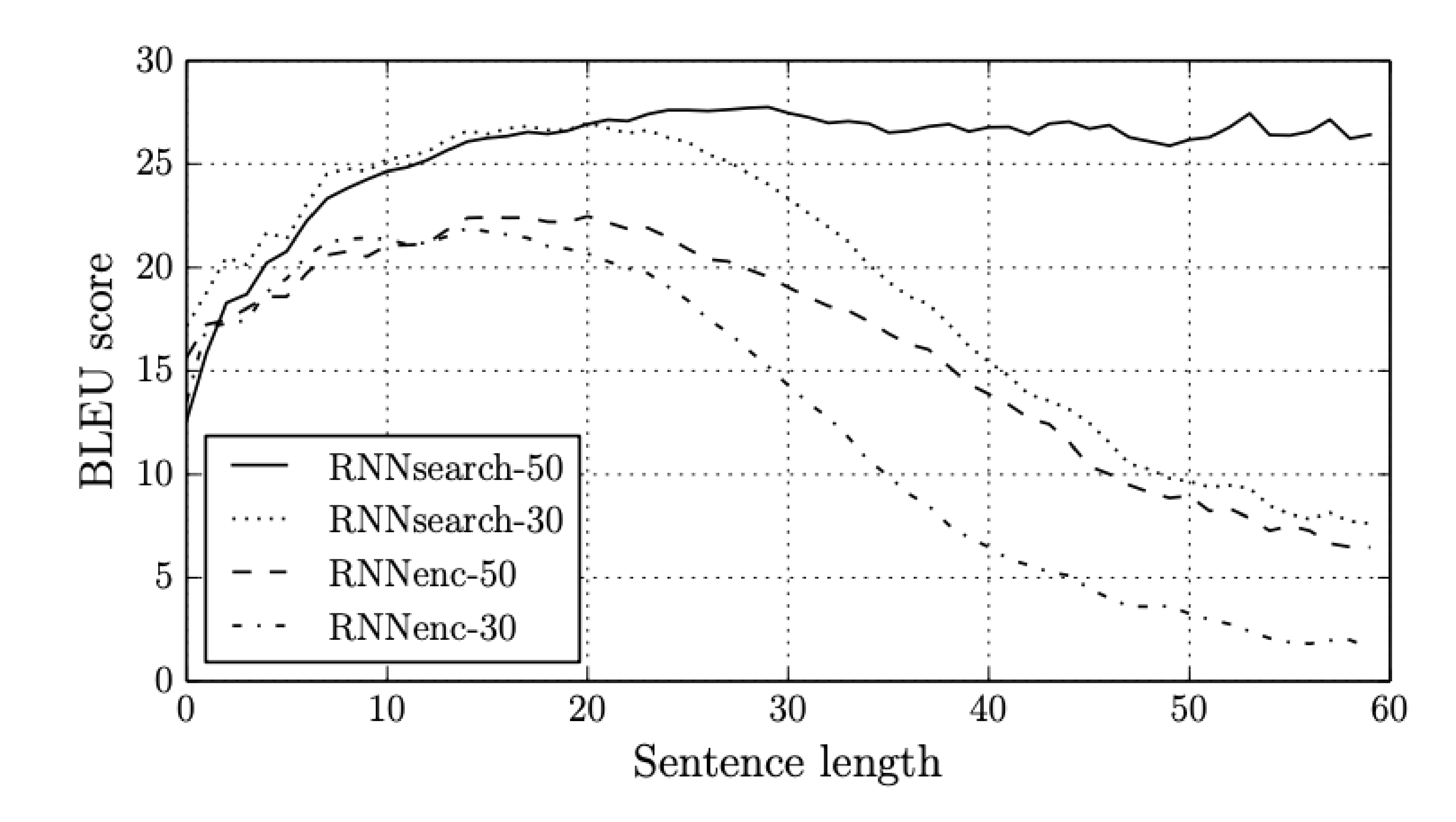
  - Prediction
    - Inferences via vector lookups

Word analogy in Word2Vec

Exercise 1

Expression	*Nearest token*
Paris - France + Italy	Rome
bigger - big + cold	colder
sushi - Japan + Germany	bratwurst
Cu - copper + gold	
Windows - Microsoft + Google	
Montreal Canadiens - Montreal + Toronto	

Attention is all you need



*Cho et al. 2014 – "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches"

*Bahdanau et al. 2015 – "Neural Machine Translation by Jointly Learning to Align and Translate"

Problem

Seq2seq context vector has trouble remembering long input sentences

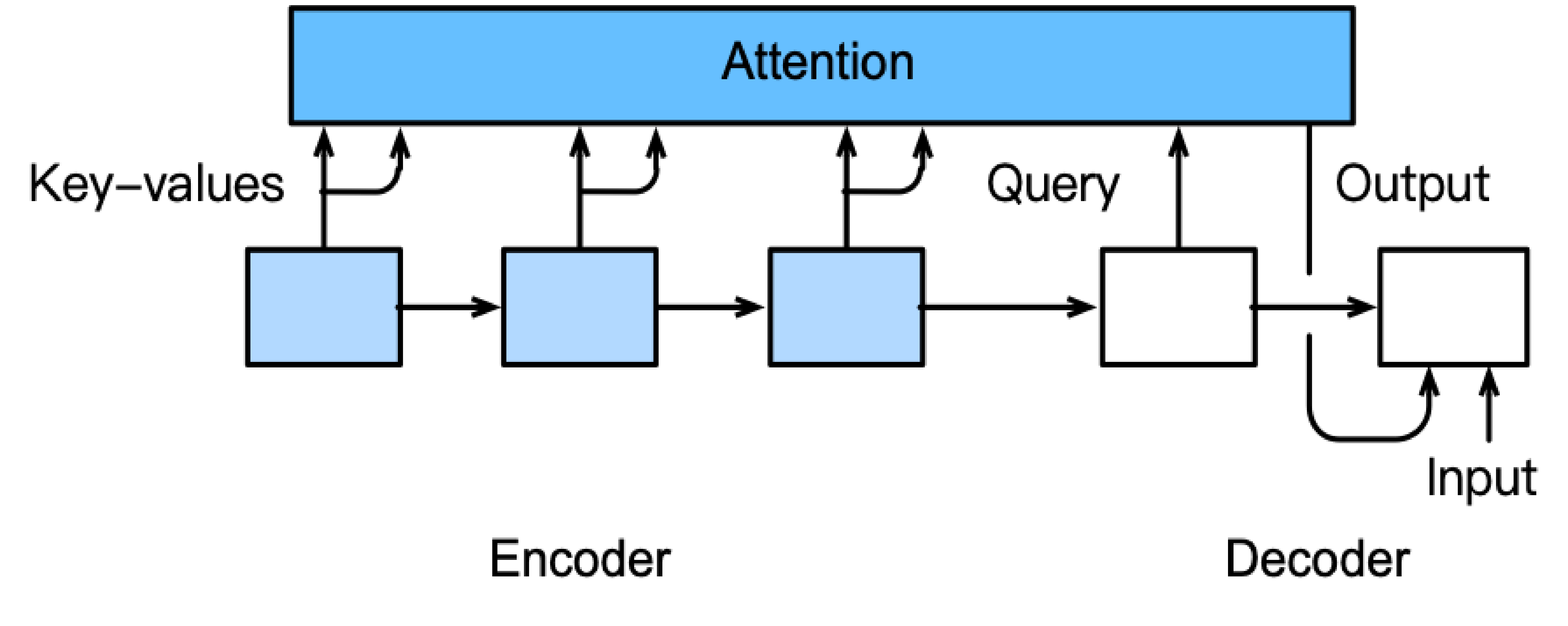
Solution

Attention* based models create learned attention vector that measures how much a single word or pixel "attends" with the other input

*Bahdanau et al. 2015 – "Neural Machine Translation by Jointly Learning to Align and Translate"

Model Architecture

Add an additional attention layer to use encoder's outputs as memory
The attention output is used as the decoder's input

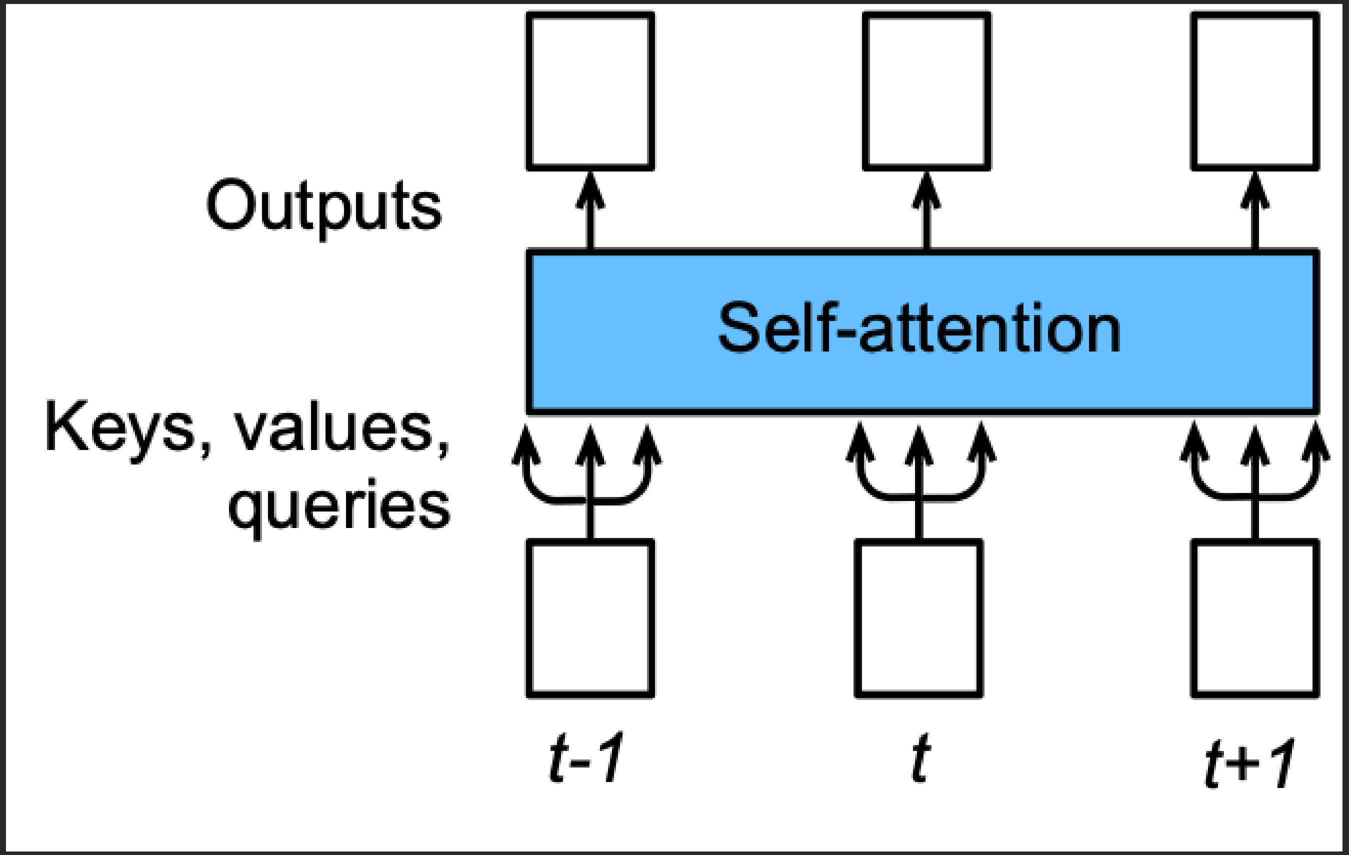


Self-attention

To generate *n* outputs with *n* inputs, we can copy each input into a key, a value and a query

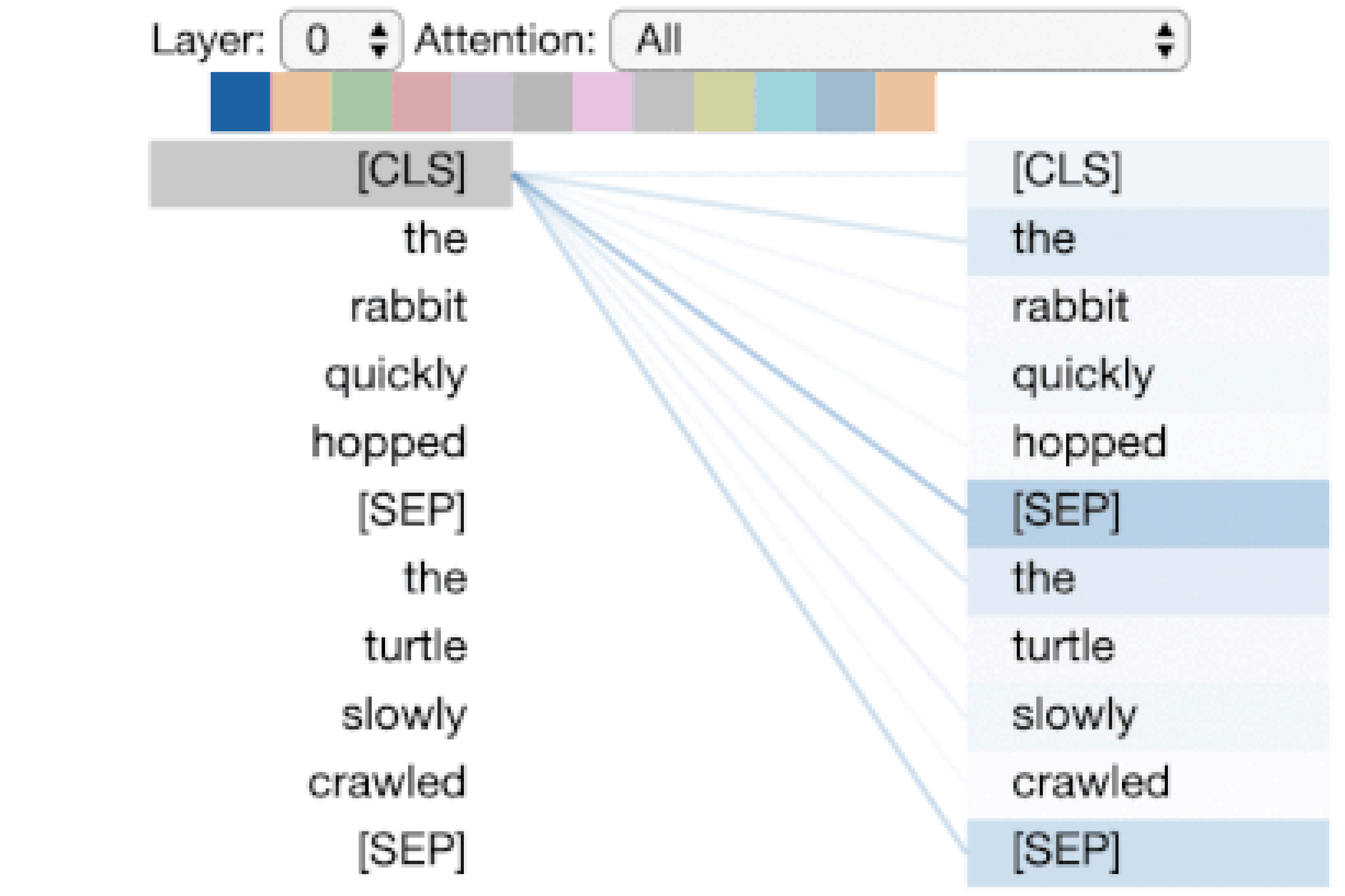
No sequential information is preserved

Run in parallel

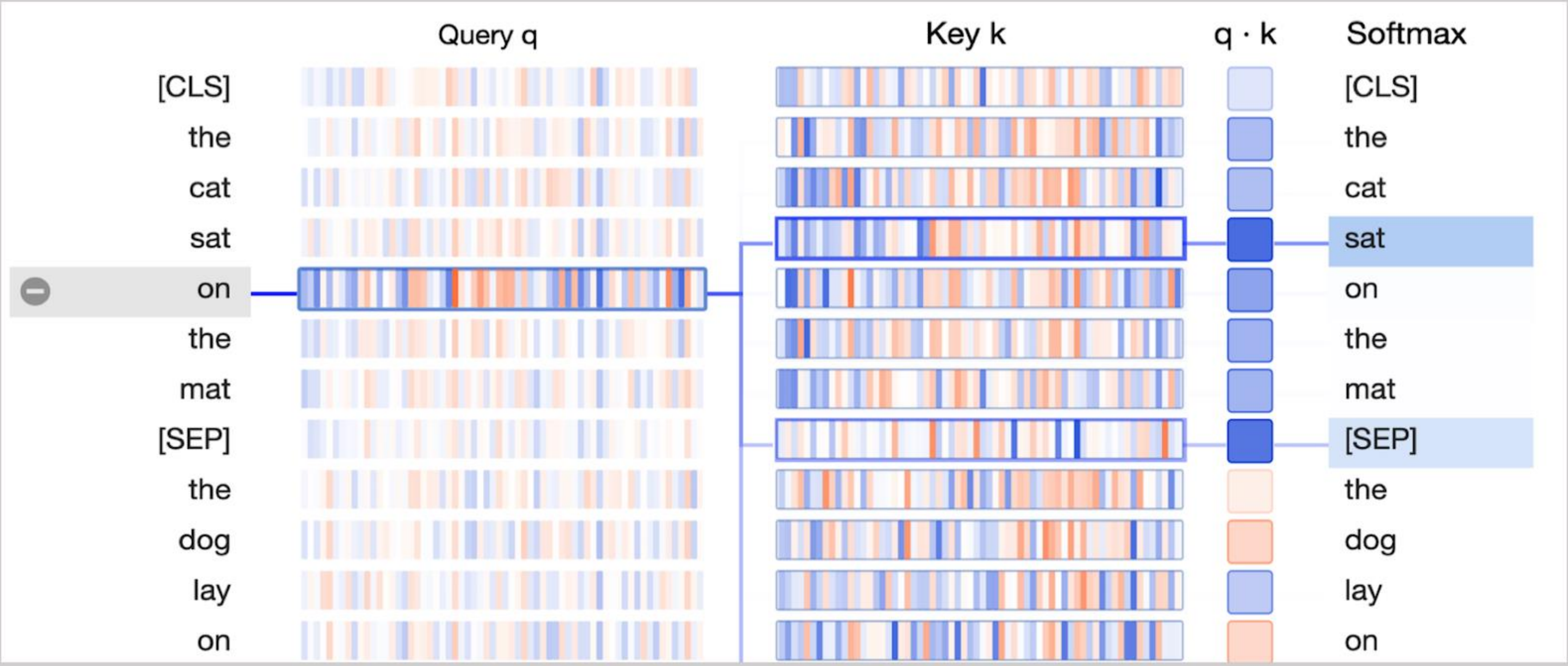


Self-attention on sentence

Self-attention:
 “The rabbit quickly hopped, the turtle slowly crawled”



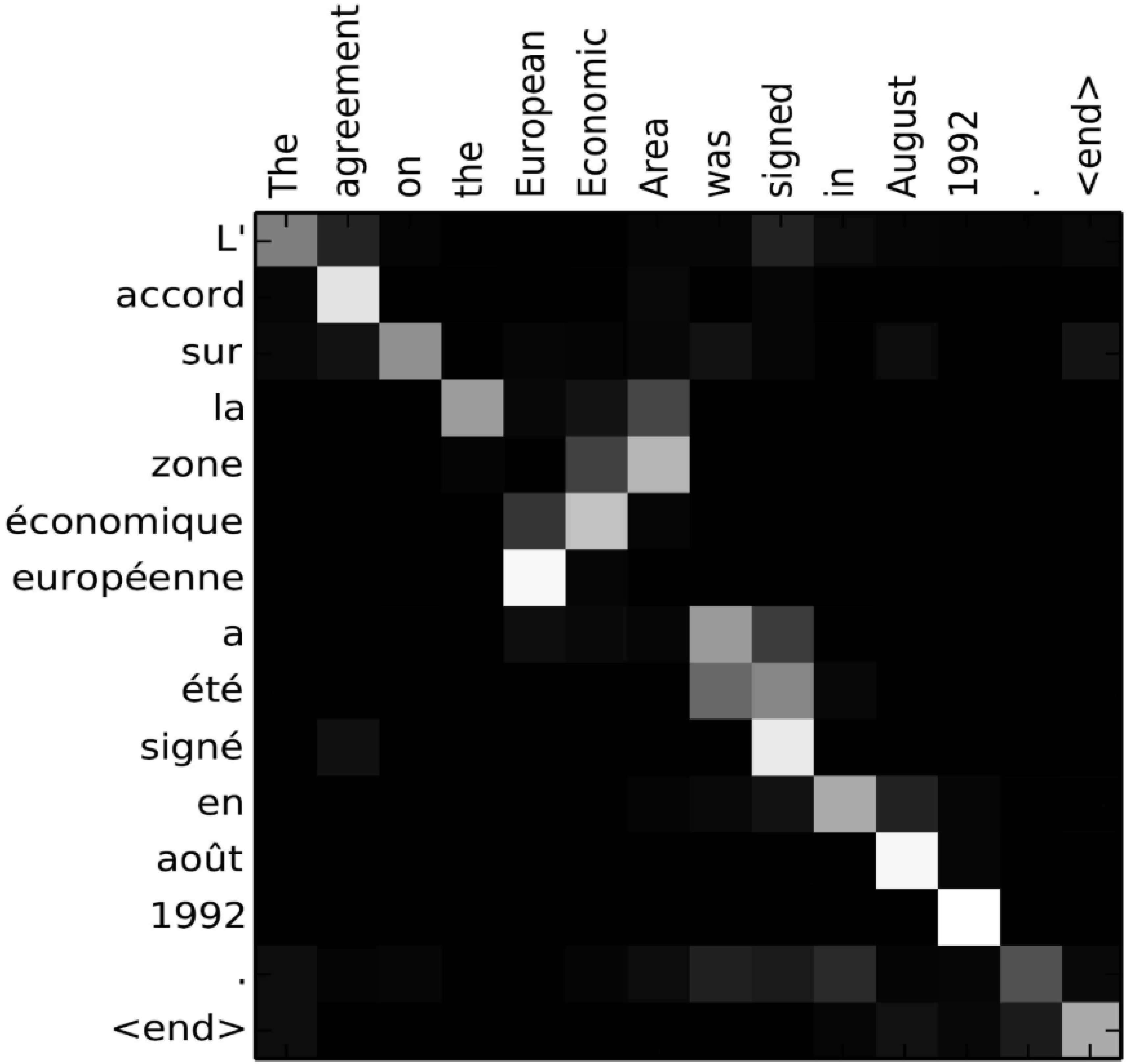
Self-attention: Q, K



Attention matrix

$\alpha_{t,i}$ - visualized,

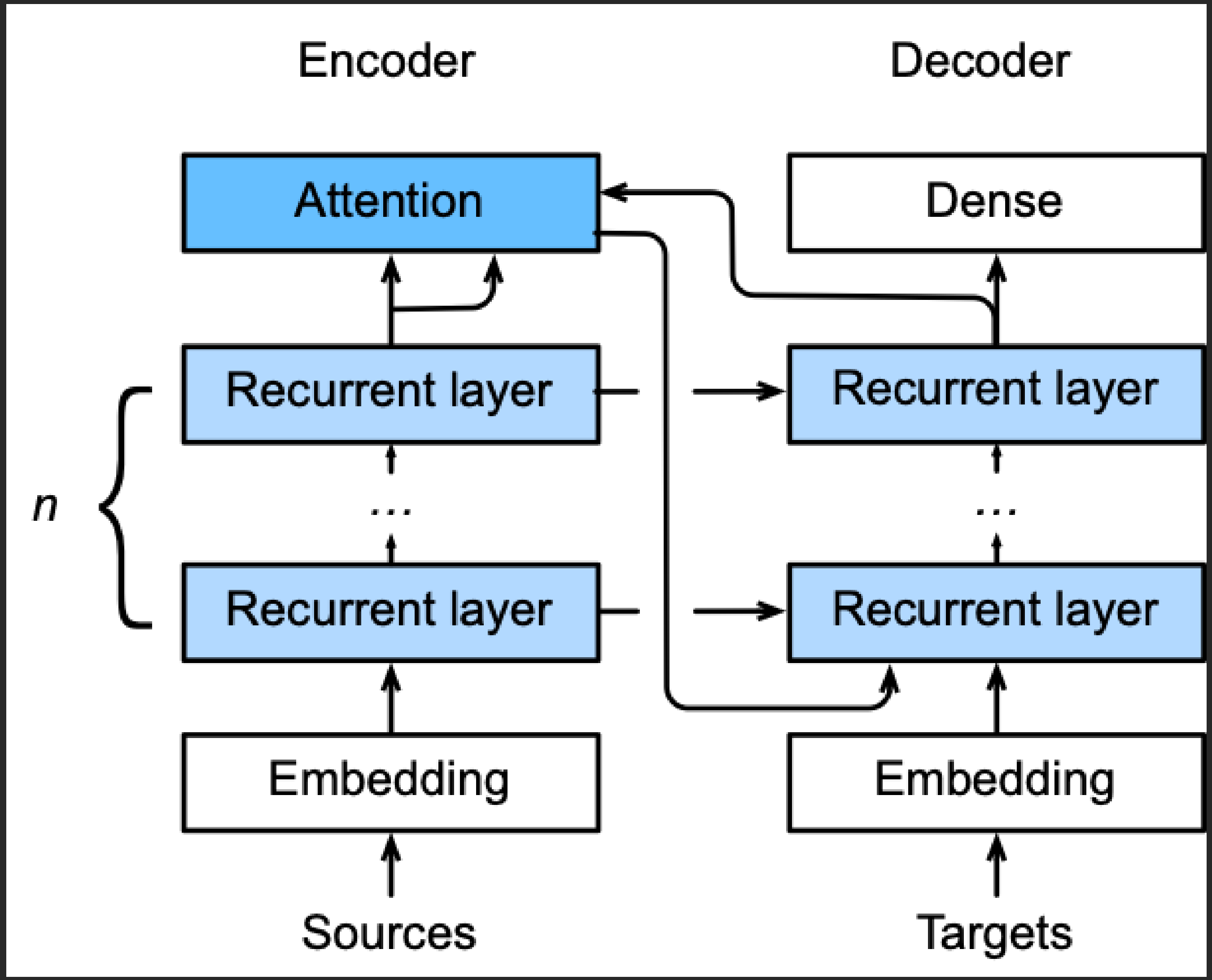
This is pairs of input at position I and output at position t based on how well they match



Encoder/Decoder Details

The output of the last recurrent layer in the encoder is used

The attention output is then concatenated with the embedding output to feed into the first recurrent layer in the decoder



Transformer

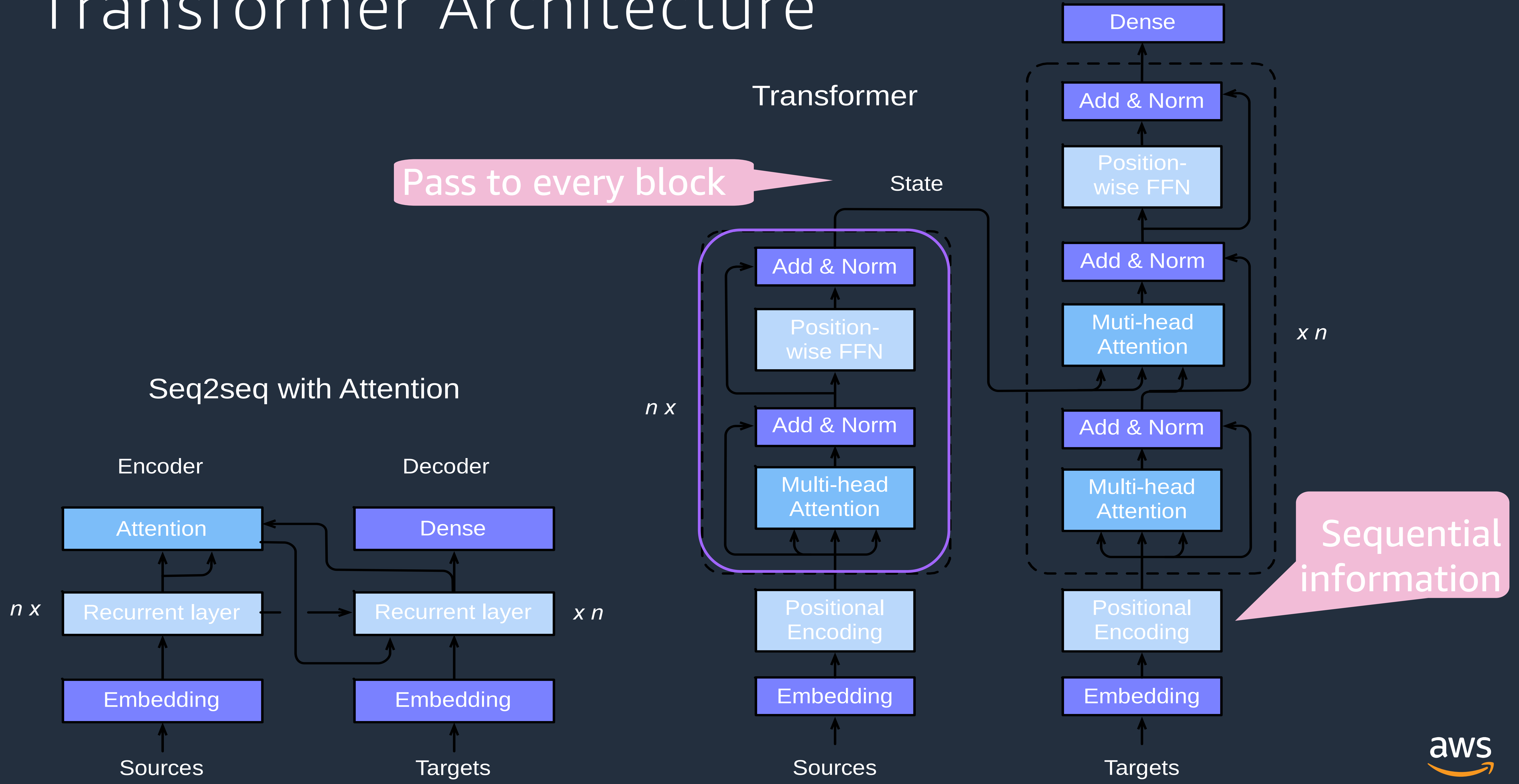
Transformer

- **CNNs**: easy to parallelize, but cannot capture the sequential dependency
- **RNNs:** able to capture the sequential information, but unable to parallelize within a sequence

- **Transformer:** combine the advantages of CNNs and RNNs

Vaswani, et al., 2017 - “attention is all you need”

Transformer Architecture



BERT –
Bidirectional Encoder Representations from Transformers

Representation learning

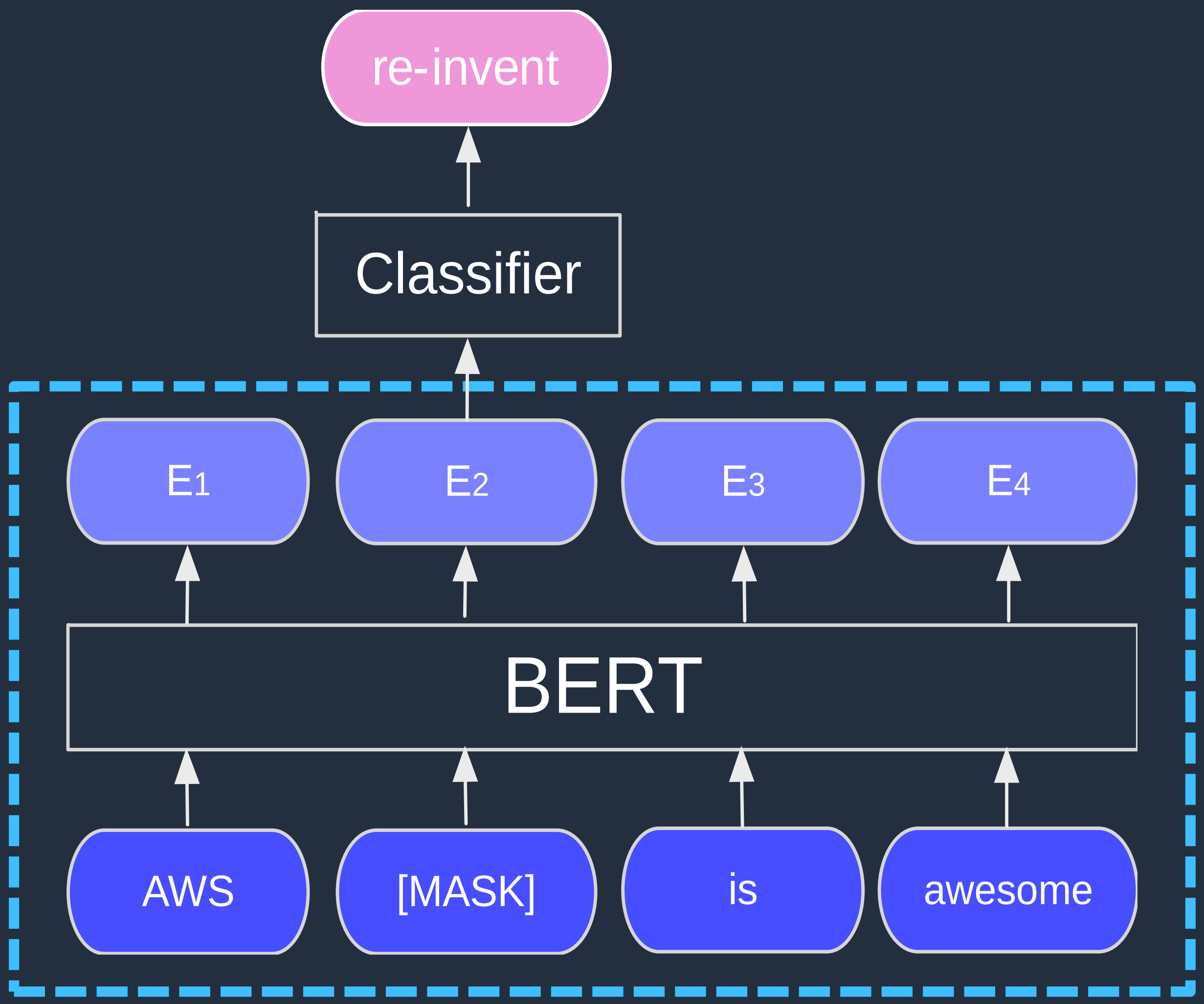
Amazon is on fire…

Representation learning with BERT

- Word embeddings
  - Vector representations of words
- Word2Vec (shallow)
- BERT (deep)
  - Bidirectional, “contextual”, deep
  - Masked language modeling

AWS [MASK] is awesome.

Outputs: P(re-invent | AWS, [MASK], is, awesome)



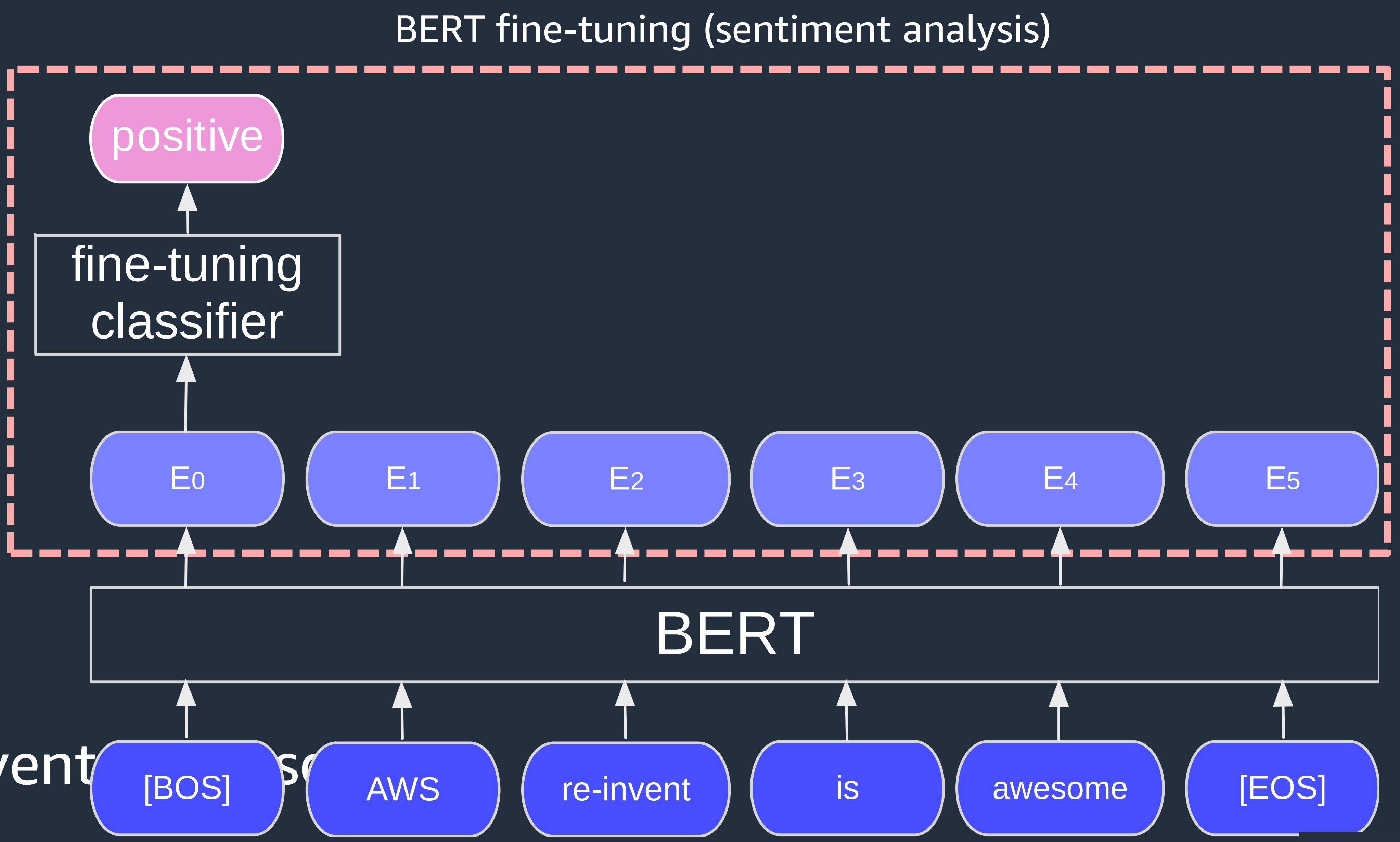
BERT Fine-tuning

Sentiment analysis

Output: positive

Embedding:

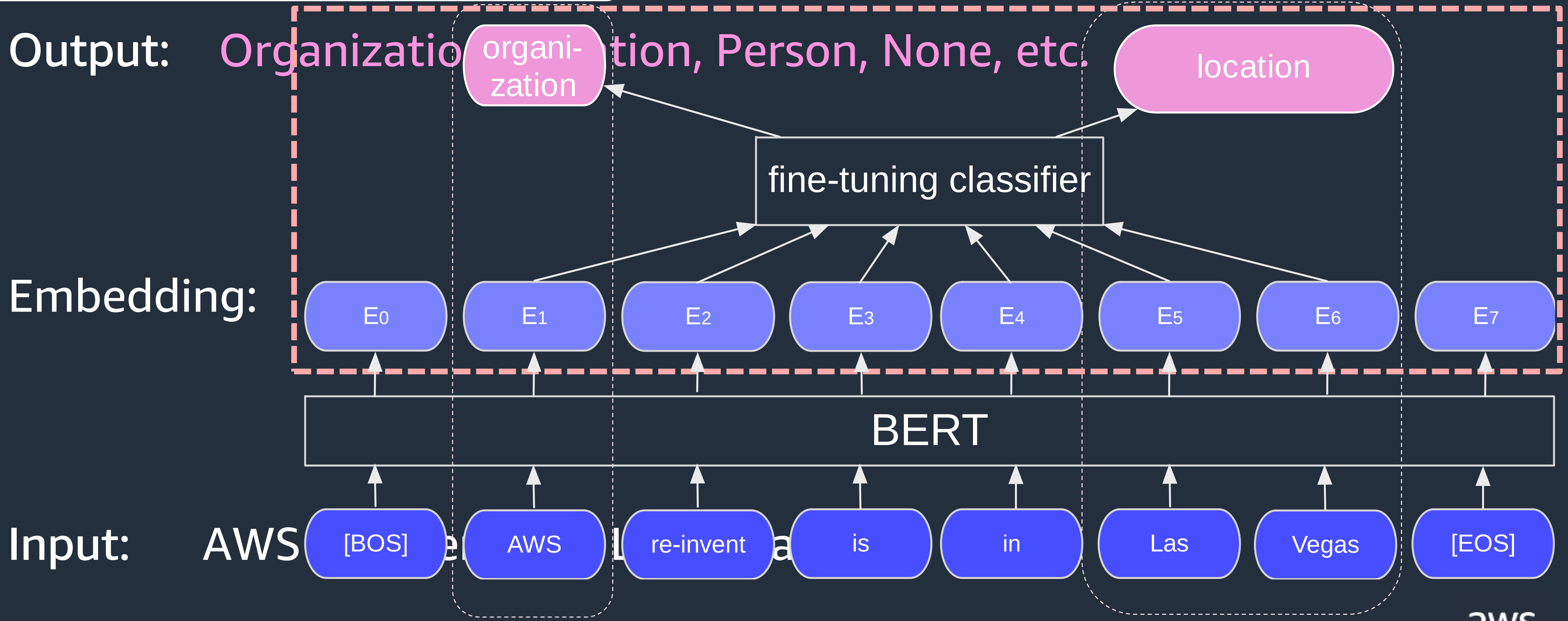
Input: AWS re-invent



BERT Fine-tuning

Name entity recognition (NER)

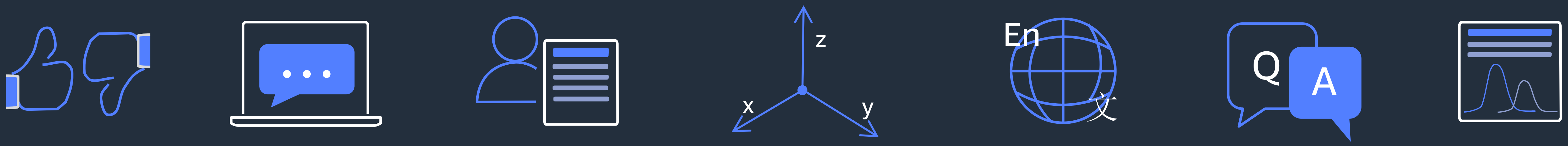
BERT fine-tuning (NER)



GluonNLP: a natural language toolkit

- State-of-the-art models
- Fast development
- Easy deployment

Multiple built-in NLP tasks



Sentiment Analysis	Text Generation	Named Entity Recognition	Representation Learning	Machine Translation	Question Answering	Language Modeling

GluonNLP: a natural language toolkit

- State-of-the-art models (pre-trained and end-to-end)
  - BERT, XLNet, GPT-2, Transformer-XL, FastText, etc

```
model, vocab = gluonnlp.model.get_model(model_name, dataset_name)
```

	Gluonnlp
Stanford sentiment treebank	**95.3 (+1.8%)**
Stanford question answering dataset	**91.0 (+2.5%)**
recognizing textual entailment	**73.6 (+7.2%)**

Further Resources

- Dive into Deep Learning http://d2l.ai/
- GluonNLP http://gluon-nlp.mxnet.io/
- GluonCV http://gluon-cv.mxnet.io/
- GluonTS https://gluon-ts.mxnet.io/
- Deep Graph Libray https://www.dgl.ai/