SAND2019-9163PE

Using Sequence-to-Sequence Models to Build Source-to-English Translations

PRESENTED BY

David Kavaler

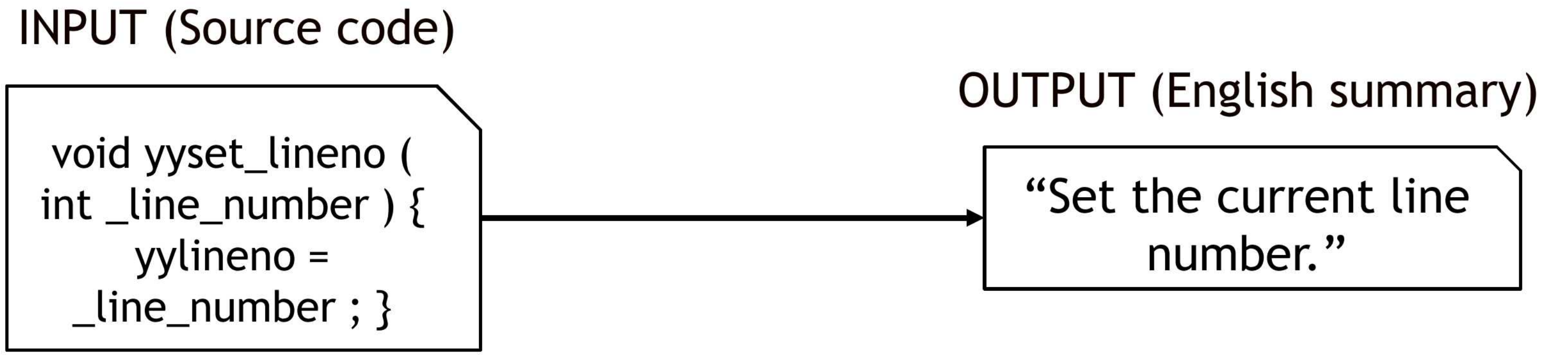
Team: Chris Harrison, Kina Kincher-Winoto, Jina Lee

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation

Can we produce English descriptions of functions from source code?

If we can, analysts can quickly identify regions of interest, find similar codes, etc.



INPUT (Source code)

```
void yyset_lineno (
int _line_number ) {
yylineno =
_line_number ; }
```

OUTPUT (English summary)

"Set the current line number."

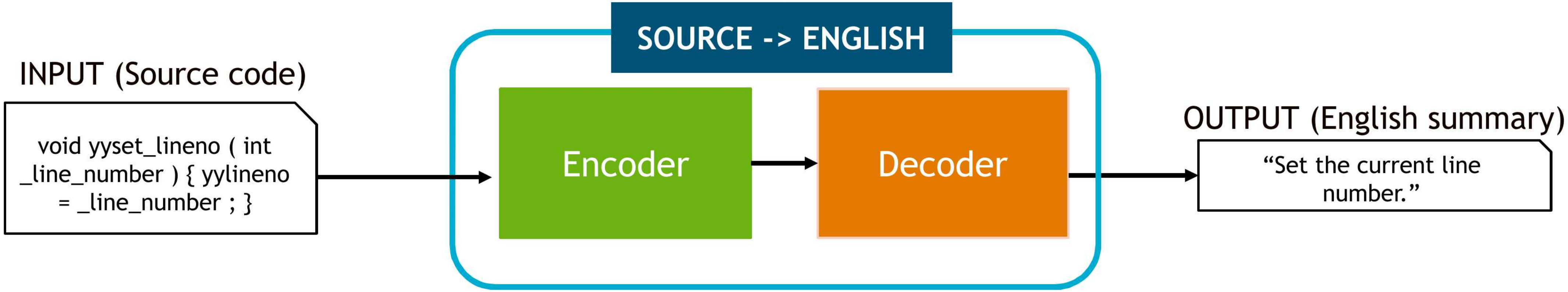
Approach

Data Idea:

Utilize comments from existing source code to use as training data set.

Algorithm Idea:

Explore deep learning methods for “translating” source codes to English summary


INPUT (Source code)
void yyset_lineno (int _line_number) { yylineno = _line_number ; }
SOURCE -> ENGLISH
Encoder
Decoder
OUTPUT (English summary)
“Set the current line number.”

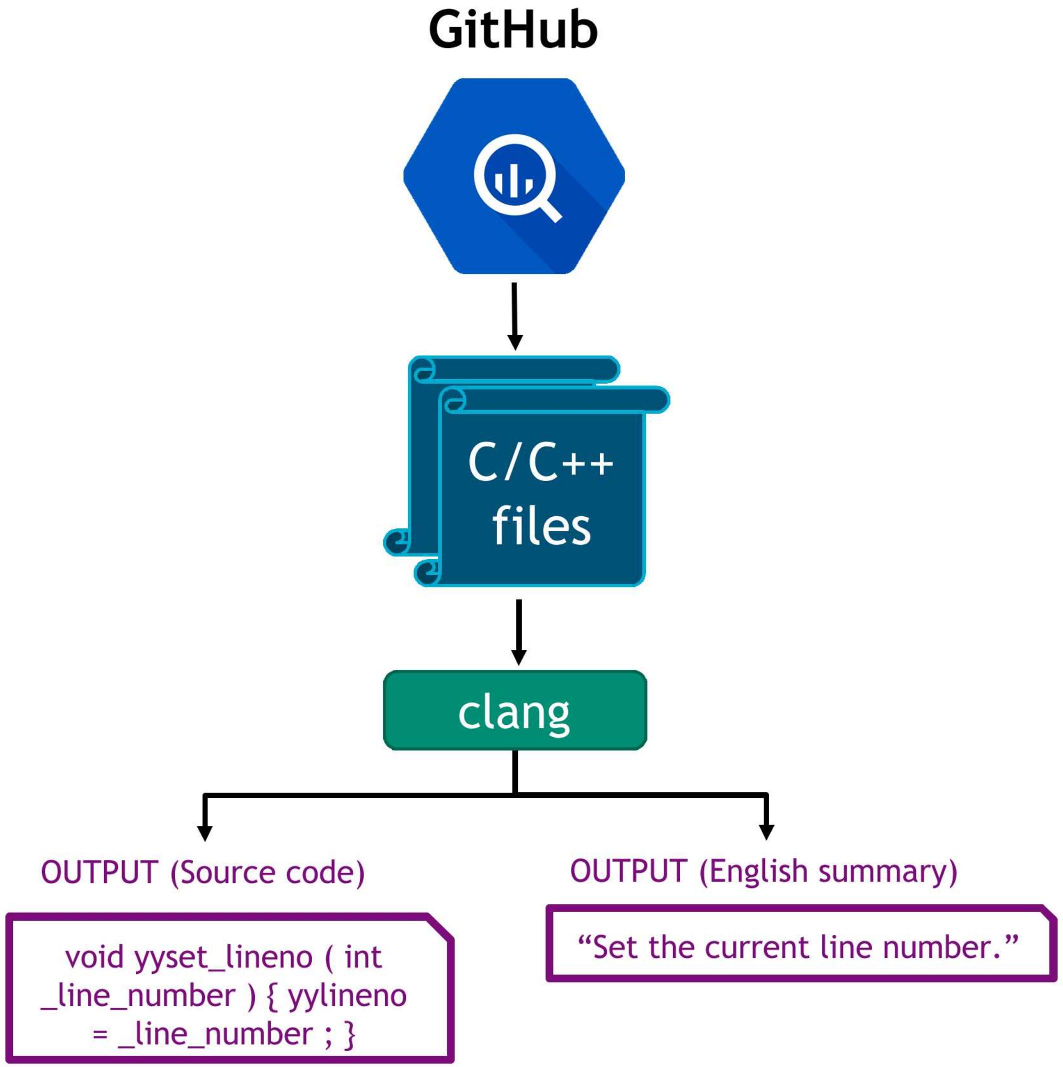
Data Acquisition

Mined GitHub C/C++ files and associated comments

- Used **`clang`** to extract C functions and associated comments
- Duplicates removed *prior* to preprocessing

Dataset

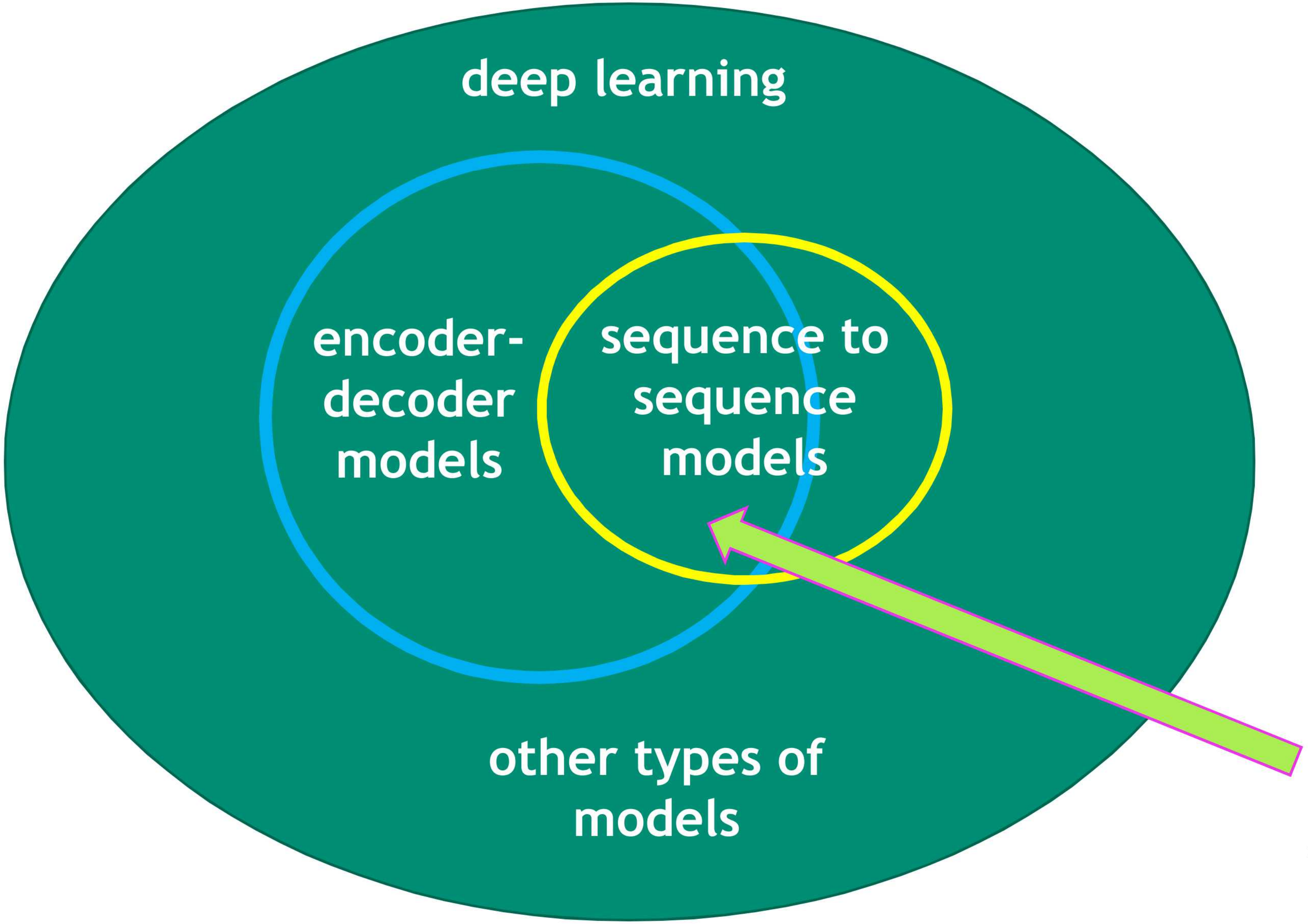
- ~1.4M train
- 338k validation
- 271k test



```
void yyset_lineno ( int _line_number ) { yylineno = _line_number ; }
```

"Set the current line number."

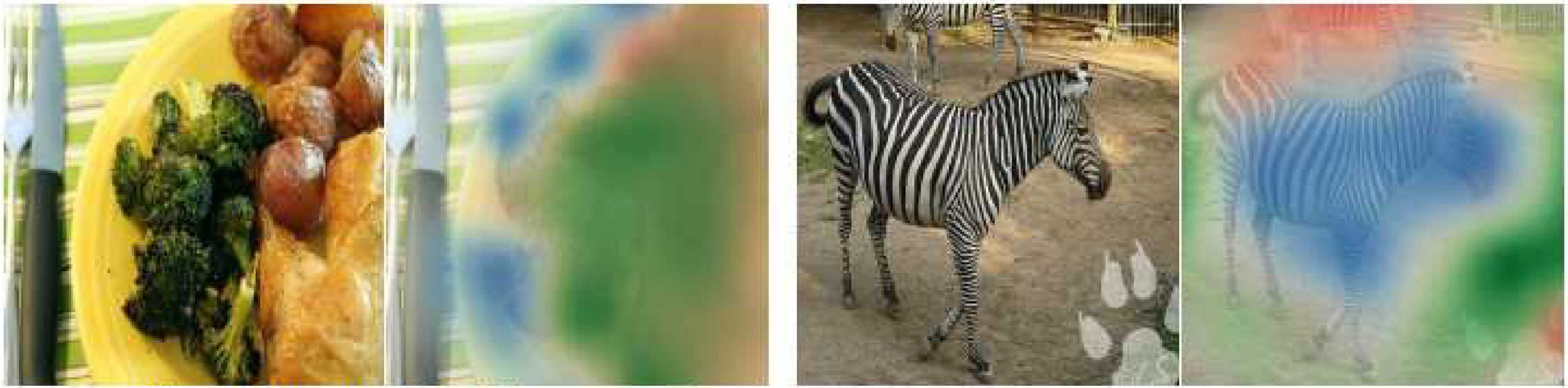
Algorithm



The model that we use sits here: **transformer model**

What is a Sequence to Sequence Model?

Our first illustrative application:

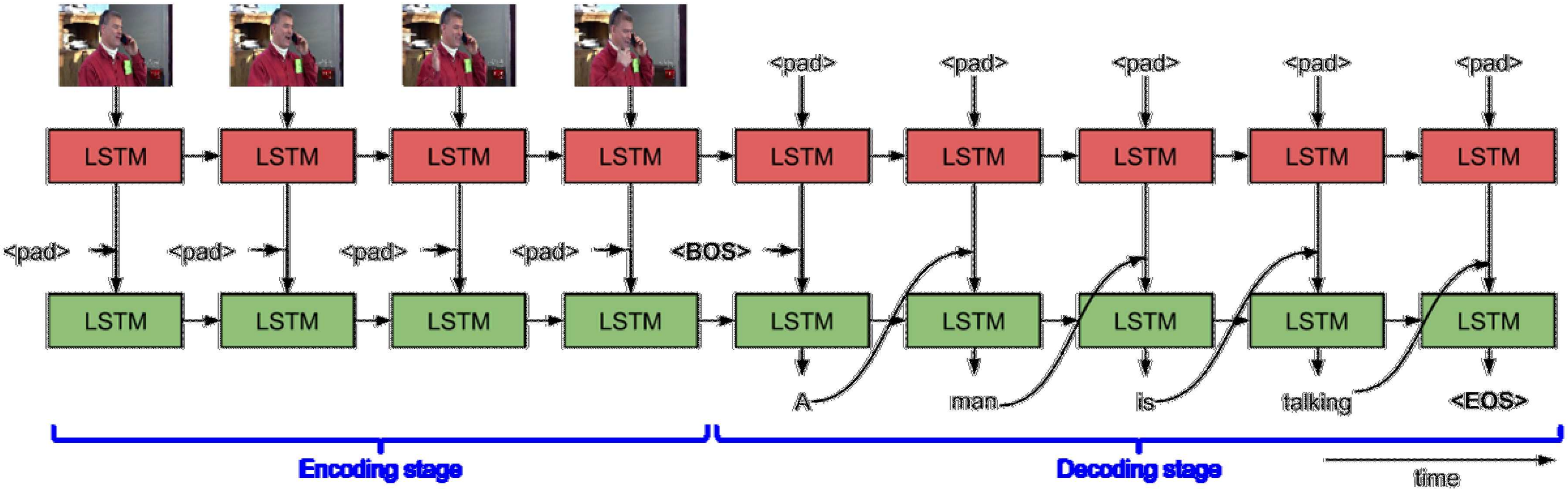


a yellow plate topped with meat and broccoli.

a zebra standing next to a zebra in a dirt field.

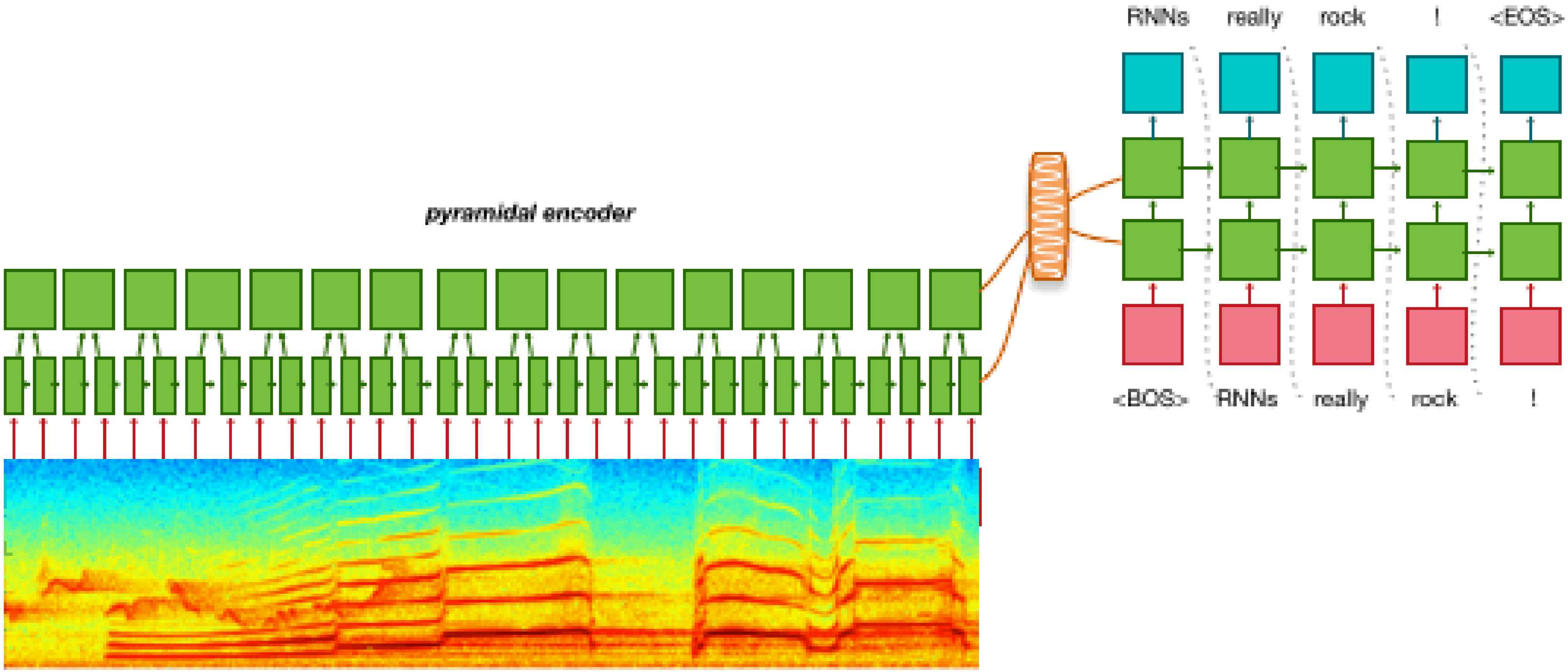
Lu et al. (2017)

Sequence to Sequence Model Applications



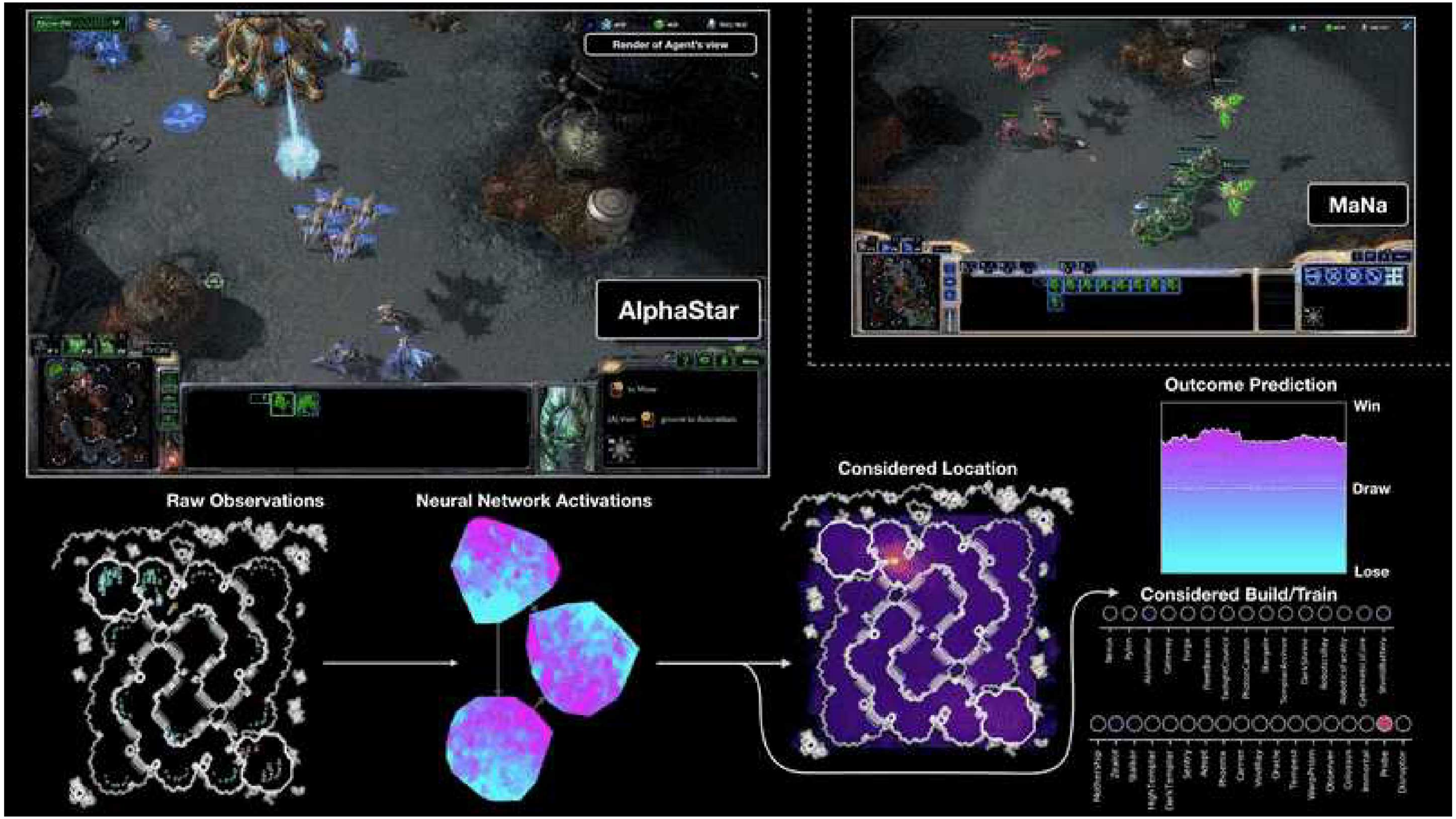
https://vsubhashini.github.io/s2vt.html

Sequence to Sequence Model Applications



http://opennmt.net/OpenNMT/applications/

Sequence to Sequence Model Applications



https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/

Sequence to Sequence Model Applications

PROMPT:

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

https://openai.com/blog/better-language-models/#sample1

Sequence to Sequence Model Applications

PROMPT:

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION:

The scientist named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

https://openai.com/blog/better-language-models/#sample1

Sequence to Sequence Model Applications

PROMPT:

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION:

The scientist named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

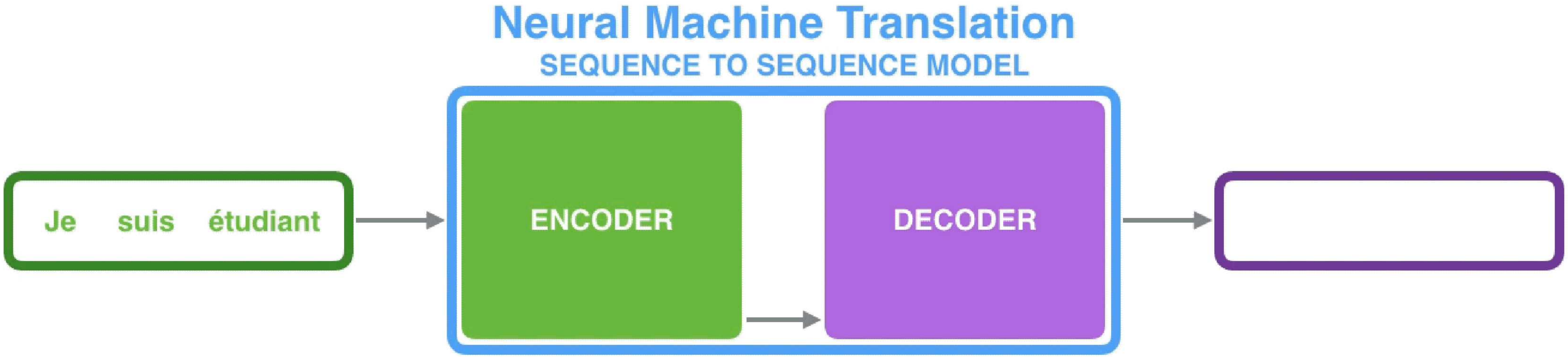
…

https://openai.com/blog/better-language-models/#sample1

Sequence to Sequence Model Applications

Deep learning models that take a sequence of items and outputs another sequence of items

- Generic input-output (sequence-sequence) format
- Successful across disciplines (machine translation, image captioning, etc)

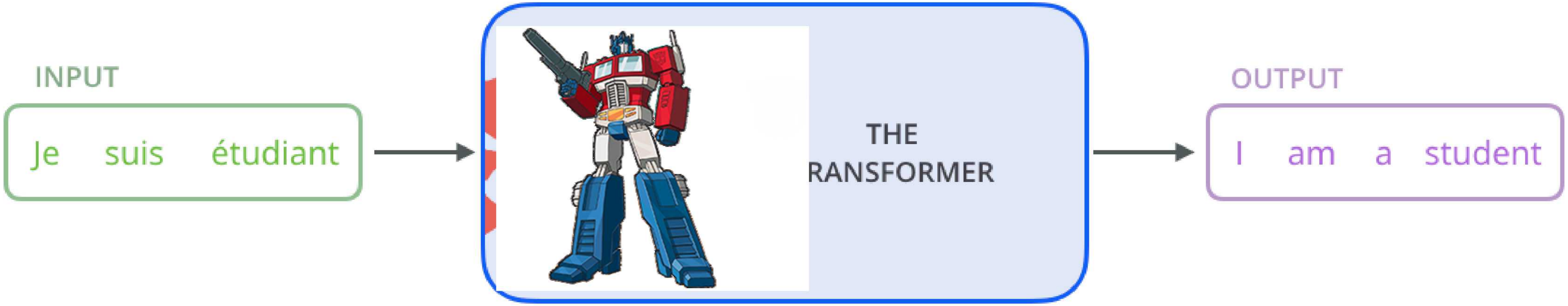


https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Sequence to Sequence Model Applications

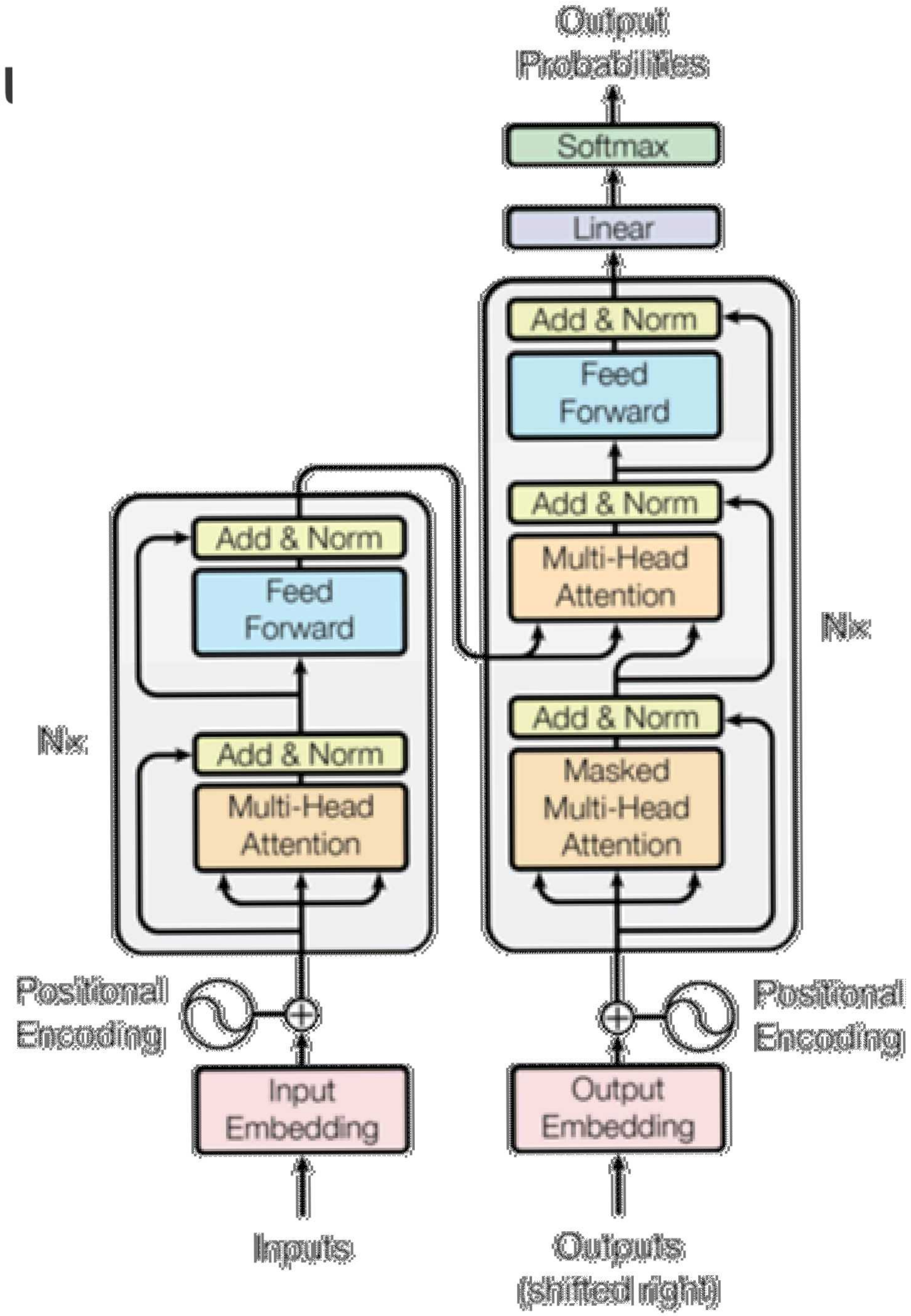
Deep learning models that take a sequence of items and outputs another sequence of items

- Generic input-output (sequence-sequence) format
- Successful across disciplines (machine translation, image captioning, etc)



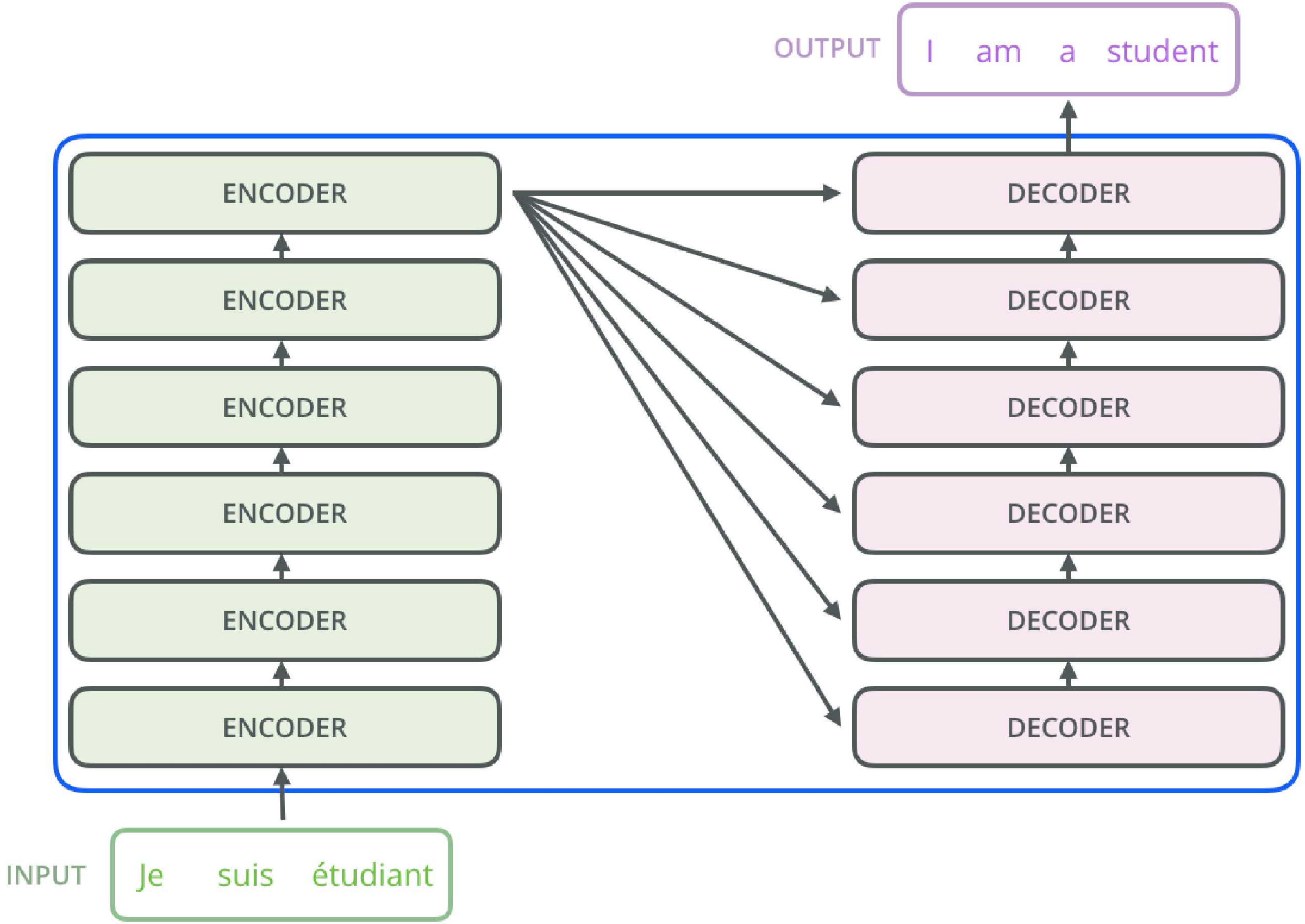
https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Transformer Architectu



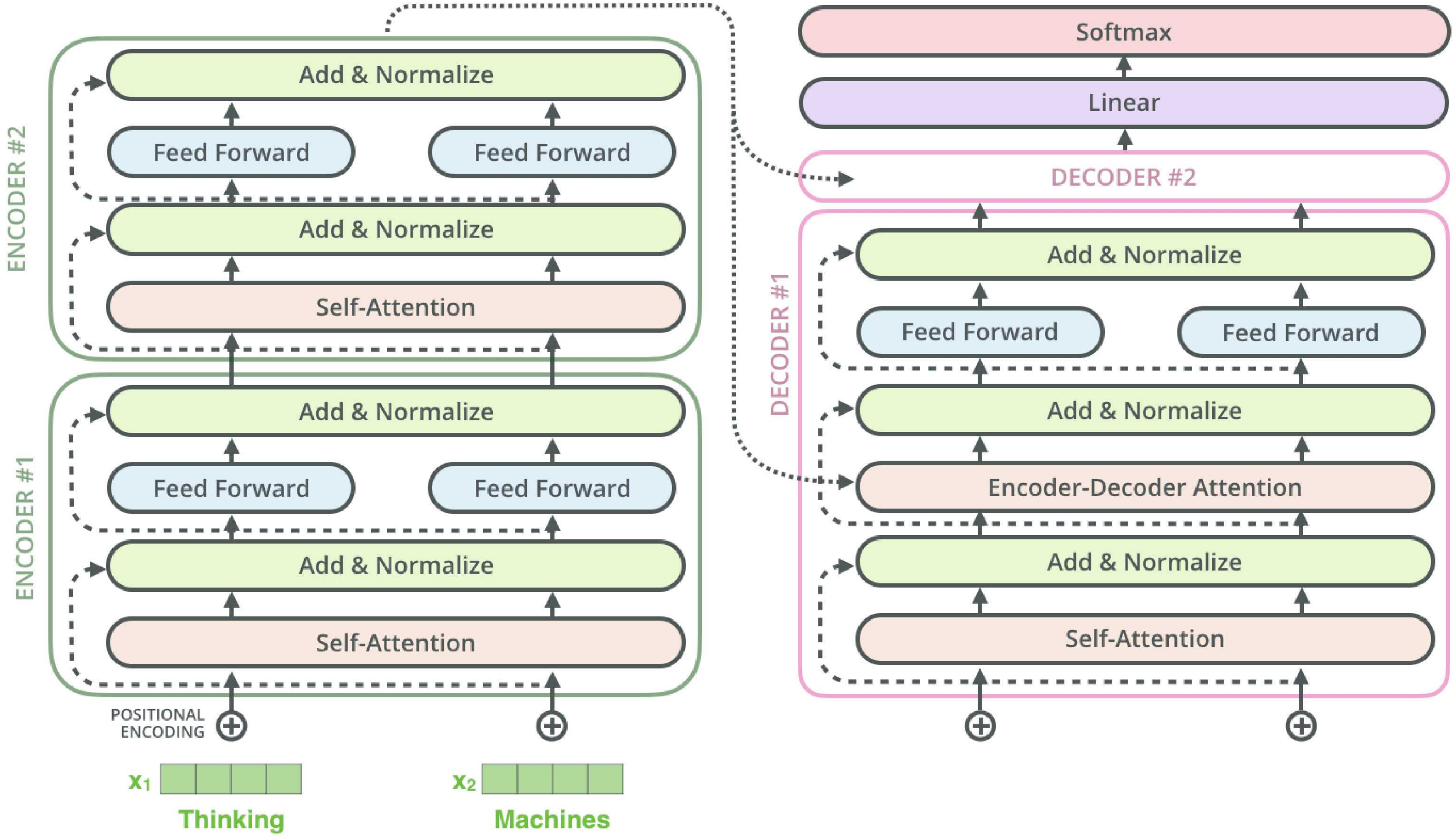
Vaswani et al. (2017)

Transformer Architecture Overview



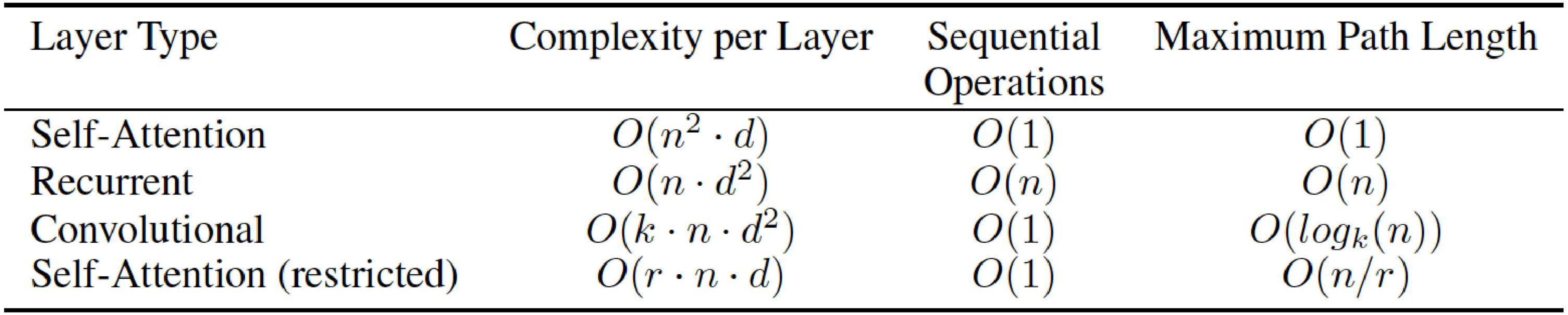
https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Transformer Architecture Overview



https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Transformer Complexity



Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Vaswani et al. (2017)

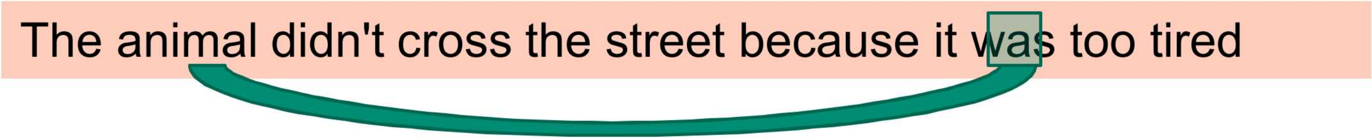
Self-Attention

The animal didn't cross the street because it was too tired

Self-Attention

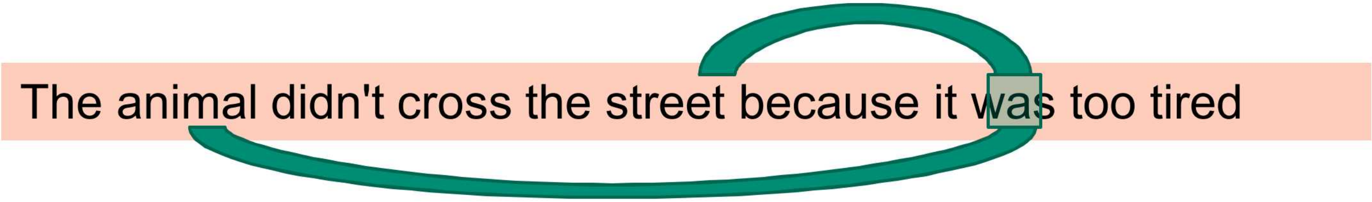
The animal didn't cross the street because it was too tired

Self-Attention



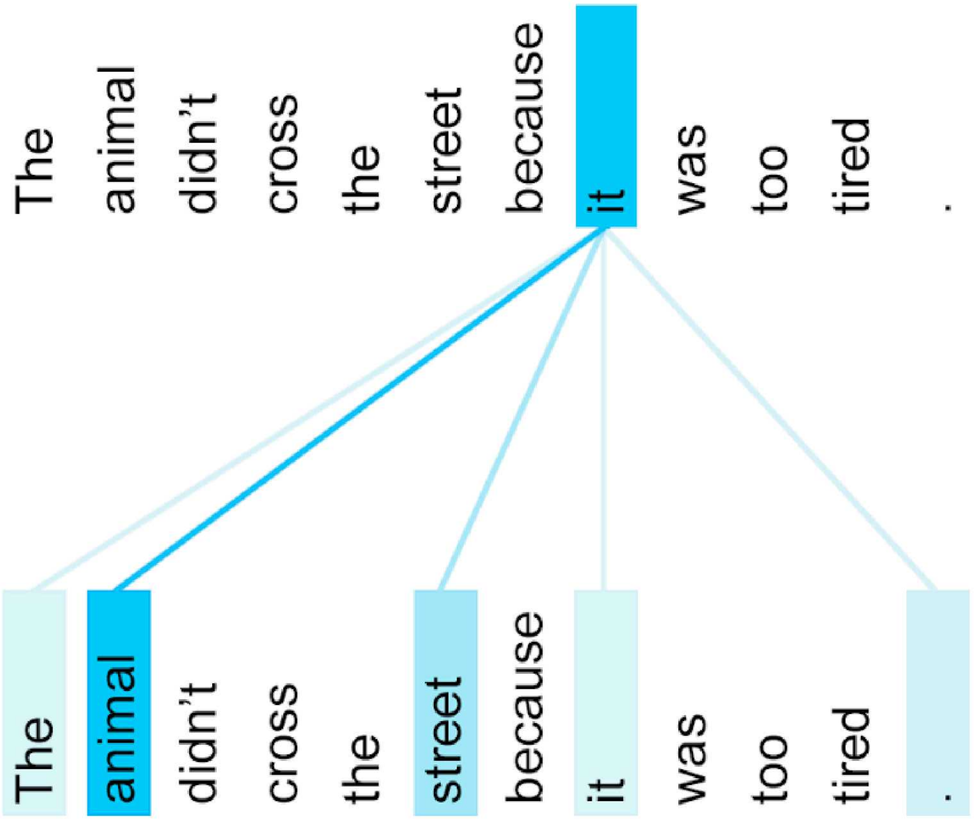
The animal didn't cross the street because it was too tired

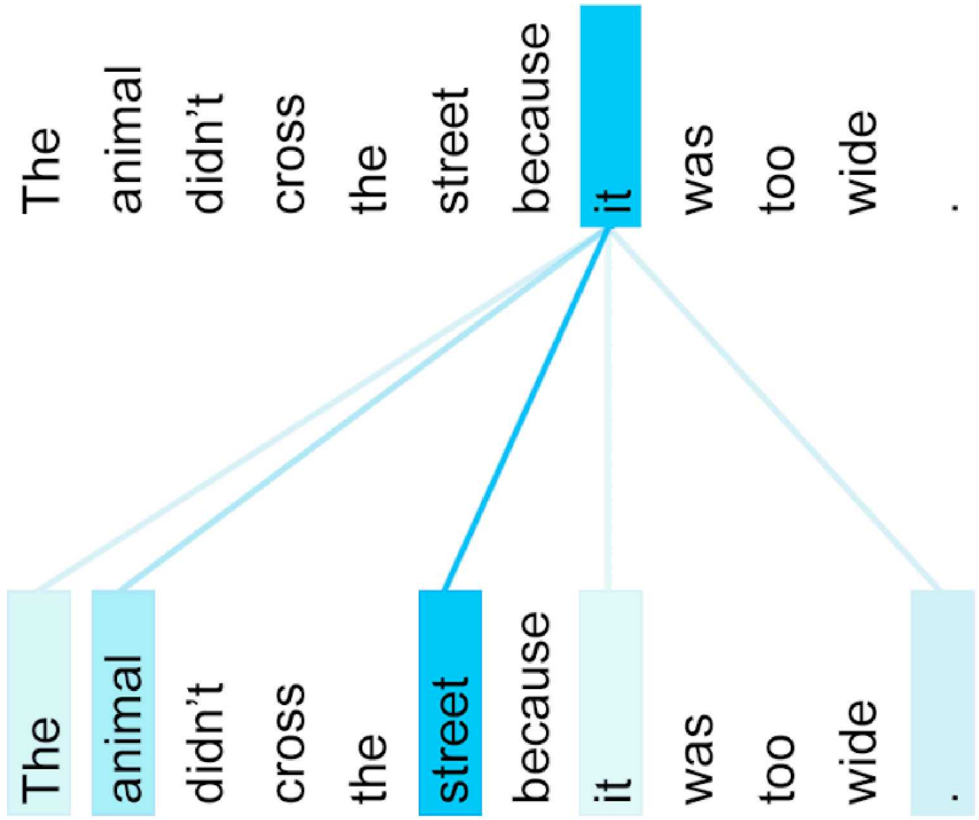
Self-Attention



The animal didn't cross the street because it was too tired

Self-Attention





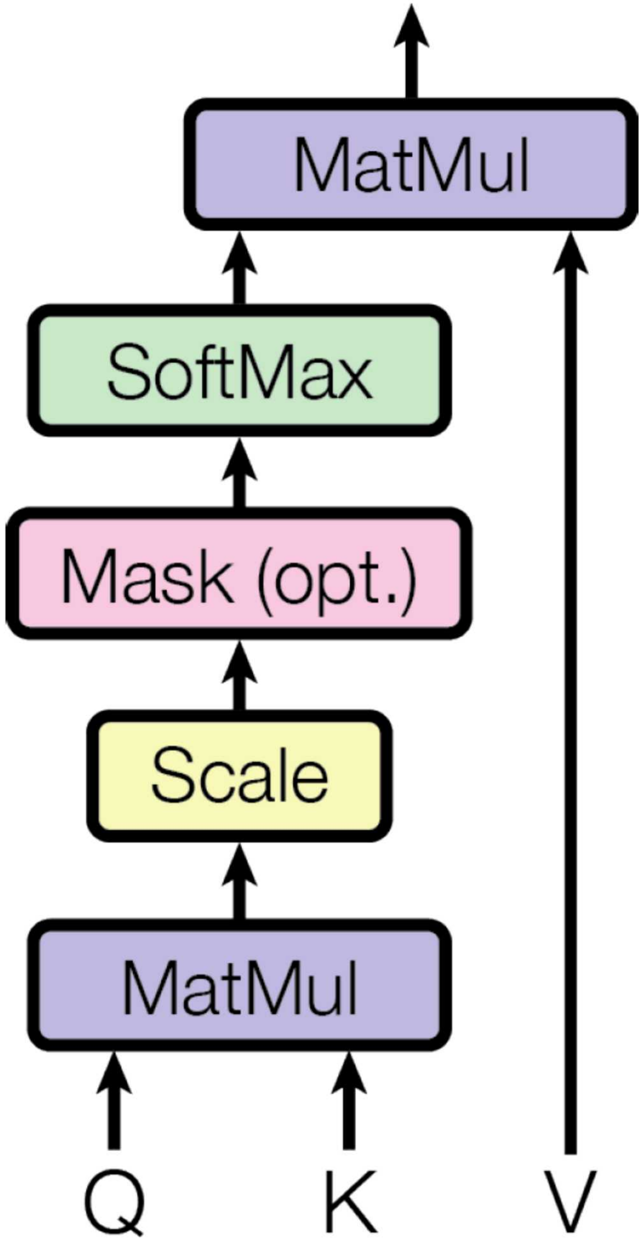
Self-Attention

Intuition:

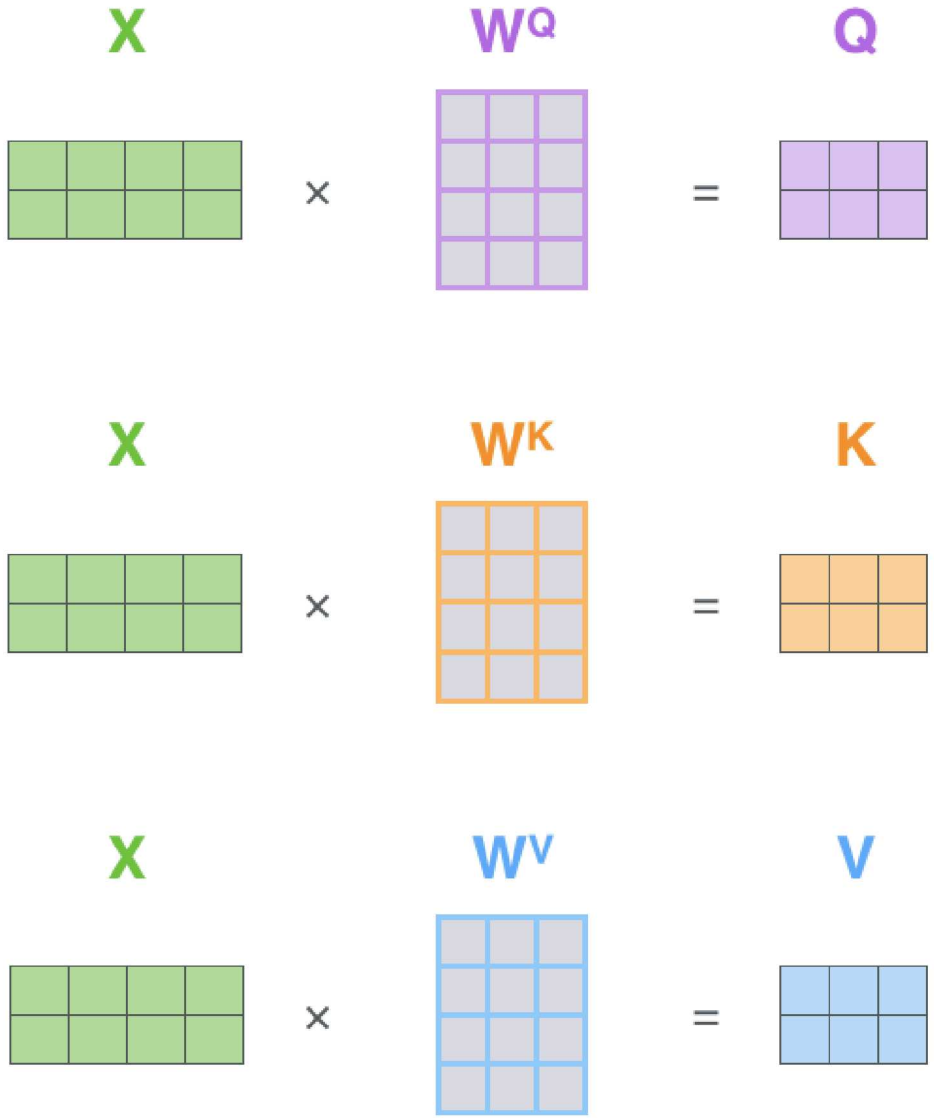
Query: current token

Key: tokens to compare with (all tokens in input sequence)

Value: output (to be scaled by softmax of Q, K operation)

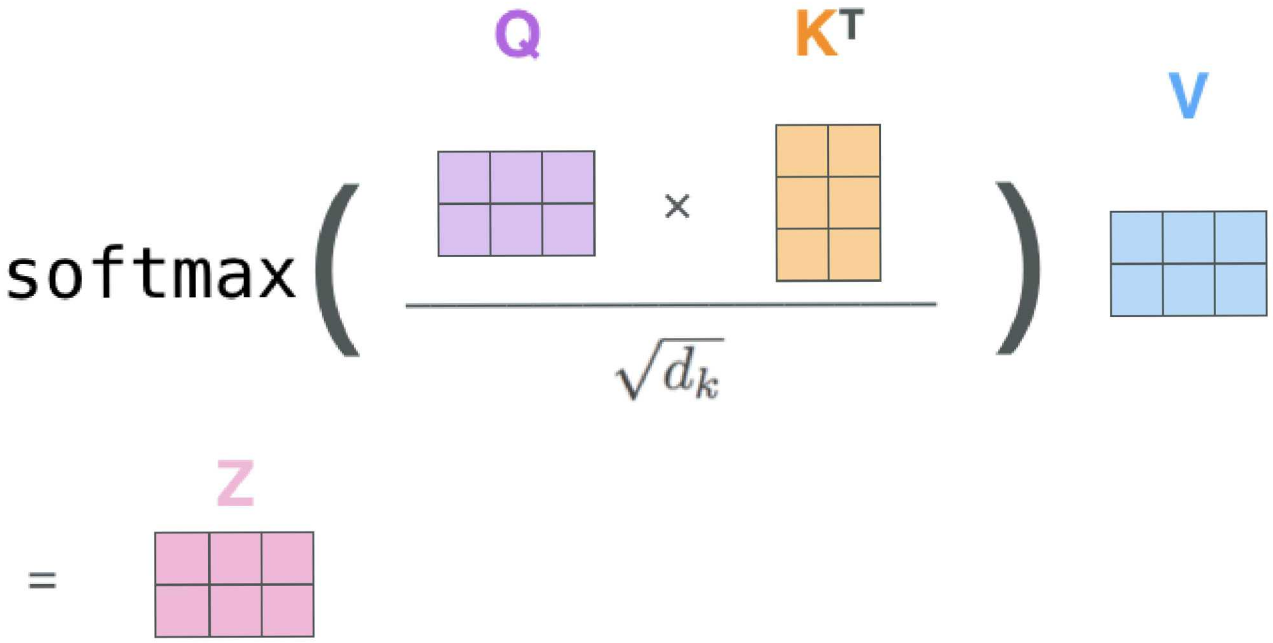


Self-Attention: Matrix Form



Self-Attention: Matrix Form

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



Back to our application…

Source-to-English Experimental Results

Data Preprocessing and Model Training

Tokenization

- Source code: split on variable names, language keywords, operators, punctuators
- Comments: replace numbers with special token, remove punctuation, normalize whitespace

Models trained

- LSTM, transformer, fconv, dynconv, transformer with back-translation
- LSTMs did not perform as well as transformers (with same number of parameters)

Evaluation: Bilingual Evaluation Underscore (BLEU)

- Popular metric for evaluating machine translation.
- Counts matching n-grams in the candidate translation to n-grams in the reference text.
  - SOTA English -> French: ~45

Our Model’s Best BLEU: 18.26 using transformer

- Training time: 16h 30m on 10 GPUs

Model 2: Source to English Model Results

Source Code (input):

```
char getField ( struct board * target , int x     , int y )
if ( x NUMBERTOKEN y NUMBERTOKEN x     target width y
target height ) return          FIELDOUTOFBOUNDS ; return *
calcFieldAddress (    target , x , y ) ;
```

English Comment (ground truth):

Gets what is on a given field of the board returns FIELDXYZ constant

Model Prediction:

Returns the value of the field at the given coordinates

Model 2: Source to English Model Results

Source Code (input):

static void makedevice (char * path , int delete) const char * devicename ; int major , minor , type , len ; int mode NUMBERTOKEN ; uidt uid NUMBERTOKEN ; gidt gid NUMBERTOKEN ; char * devmajmin path strlen (path) ; *...rest of code omitted for slide brevity*

English Comment (ground truth):

mknod in dev based on a path like sysblockhdahda1

Model Prediction:

mknod in dev based on a path like sysblockhdahdahdahdahdahdahdahdahdahdahdahdahda1 based on a dev based on a dev based on a path like sysblockhdahdahda1

Takeaways

You should try Transformers as a model!

We need more data.

If you have anything that might fit the bill, talk to us.

If you have any ideas on:

- How to use sequence-to-sequence models on highly structured data
- Other approaches for automated code descriptions.

Talk to us.

Contact us:

David Kavaler
dkavale@sandia.gov

Jina Lee
jlee3@sandia.gov

Kina Kincher-Winoto
kwinoto@sandia.gov