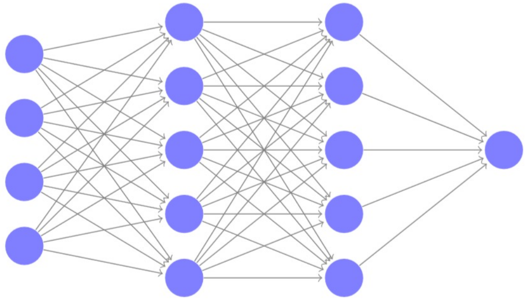
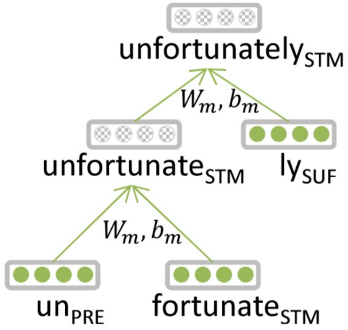
Seq2Seq Models

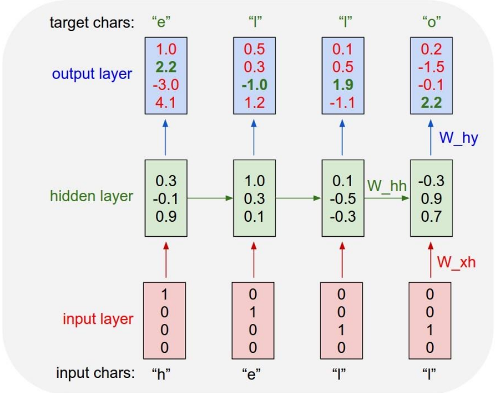
AUSTIN BLODGETT

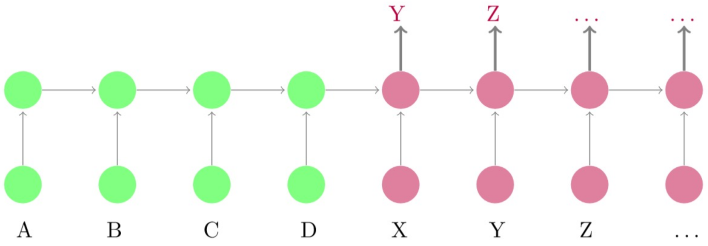
a quick review

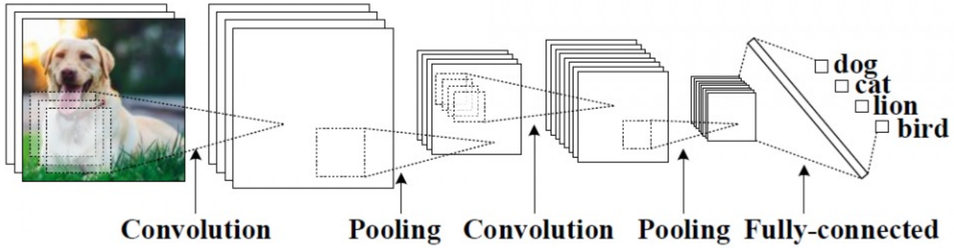
a family of algorithms











NN Task	Example Input	Example Output
Binary classification	features	+/-
Multiclass classification	features	decl, imper, inter, …
Sequence	sentence	POS tags
Sequence to Sequence	(English) sentence	(Spanish) sentence
Tree/Graph Parsing	sentence	dependency tree or AMR parse

NN Task	Example Input	Example Output
Binary classification	features	+/-
Multiclass classification	features	decl, imper, inter, …
Sequence	sentence	POS tags
Sequence to Sequence	(English) sentence	(Spanish) sentence
Tree/Graph Parsing	sentence	dependency tree or AMR parse

Seq2Seq Tasks

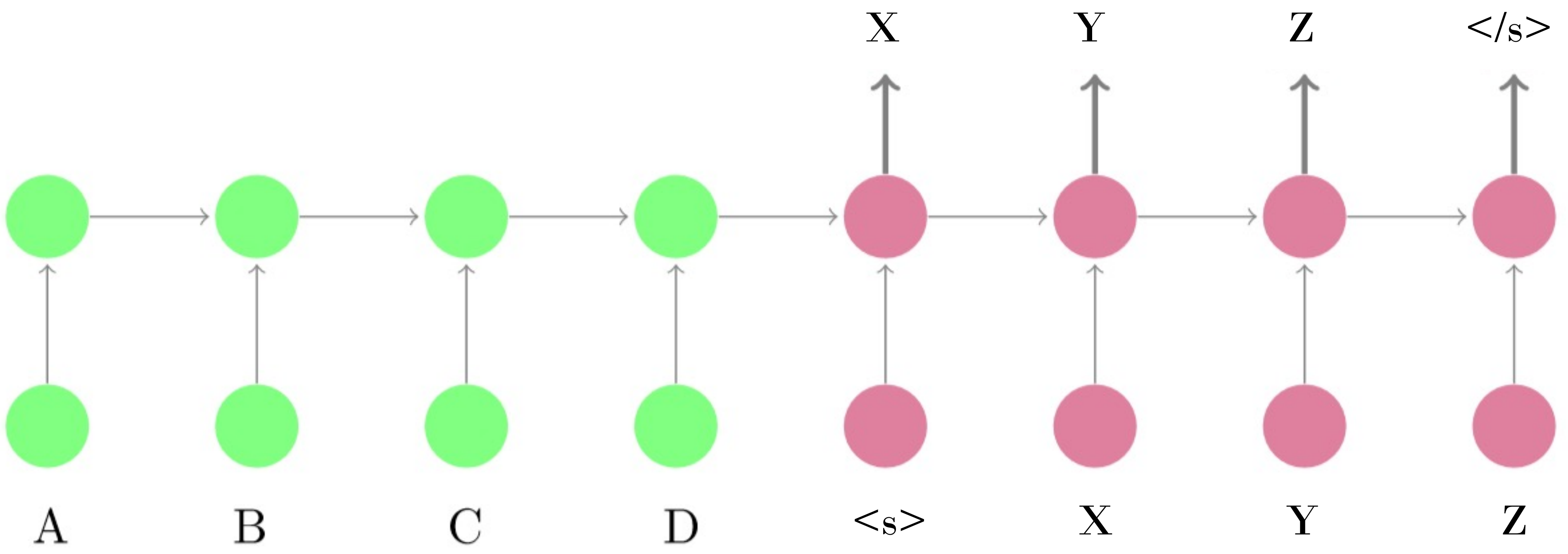
- Tasks
  - Machine Translation
  - Automatic Dialogue
  - Question Answering
  - Document Summarization
  - (Some) Semantic Parsing

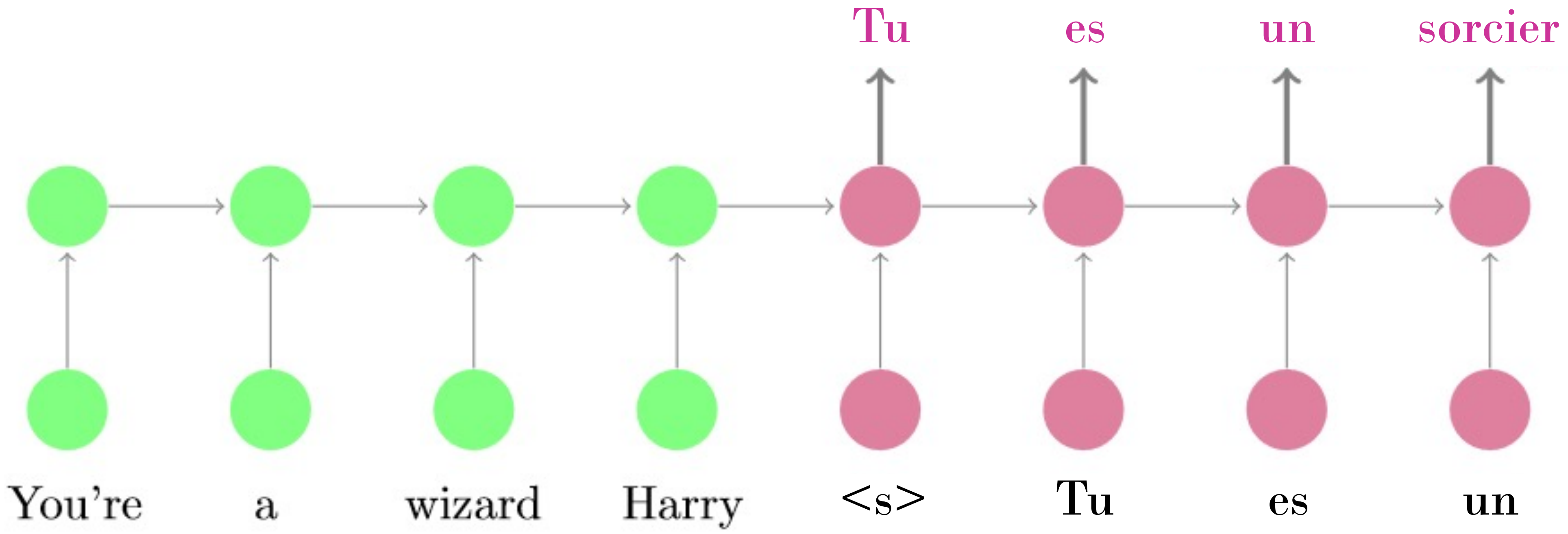
Encoder-Decoder models

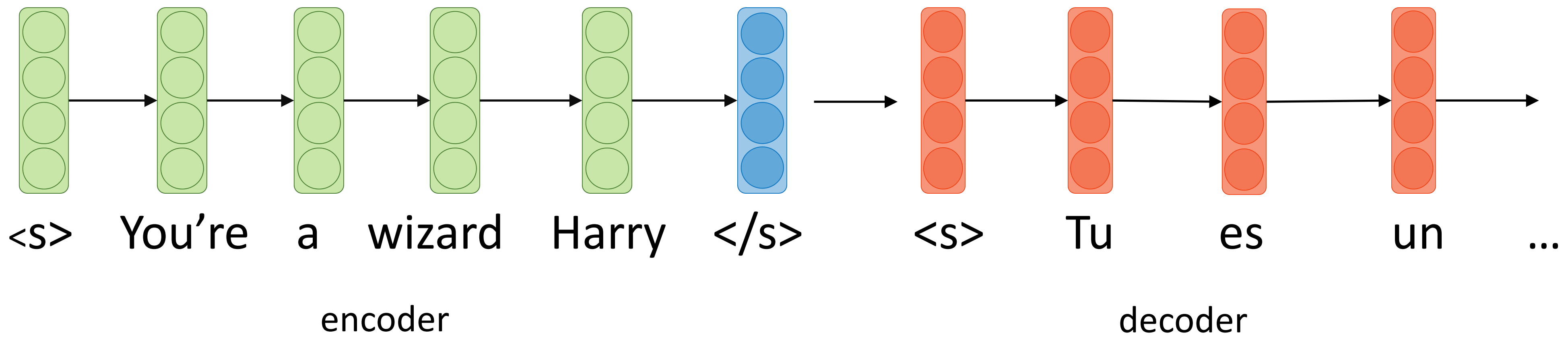
- **Encoder-Decoder model** (also **Seq2Seq**) – Take a sequence as input and predict a sequence as output
- *Input and Output may be different lengths*
- Encoder (*RNN*) models input, Decoder (*RNN*) models output

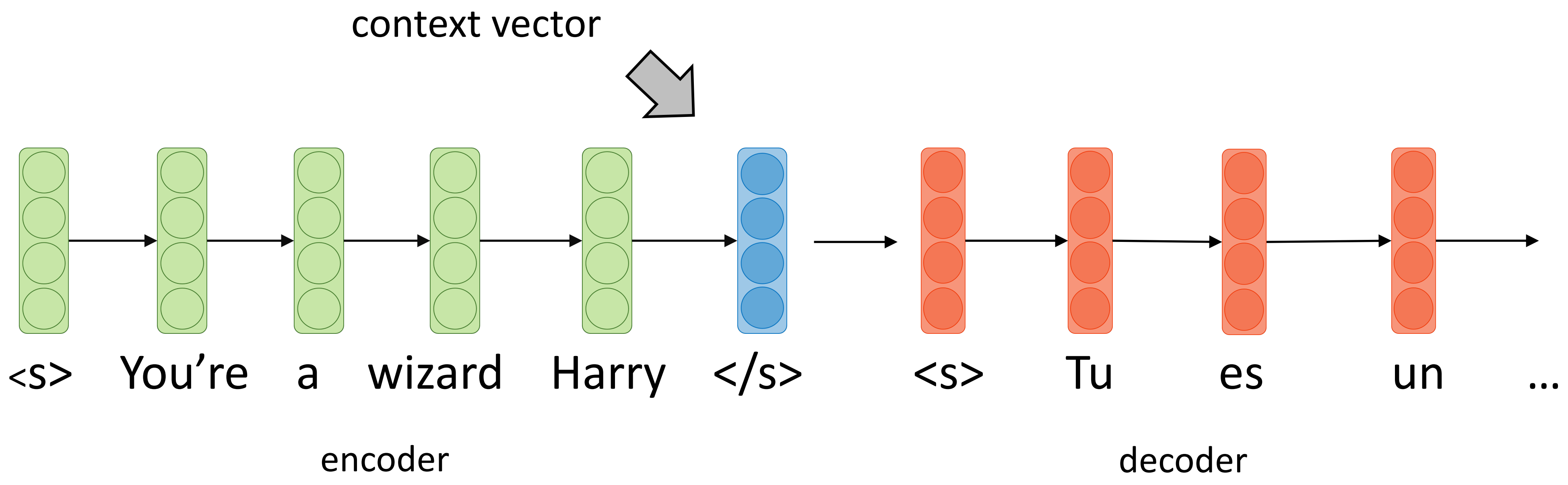
Sutskever, I., Vinyals, O., & Le, Q. V. (2014). **Sequence to sequence learning with neural networks**. In *Advances in neural information processing system.*

Cho, K., et al. (2014). **Learning phrase representations using RNN encoder-decoder for statistical machine translation**.

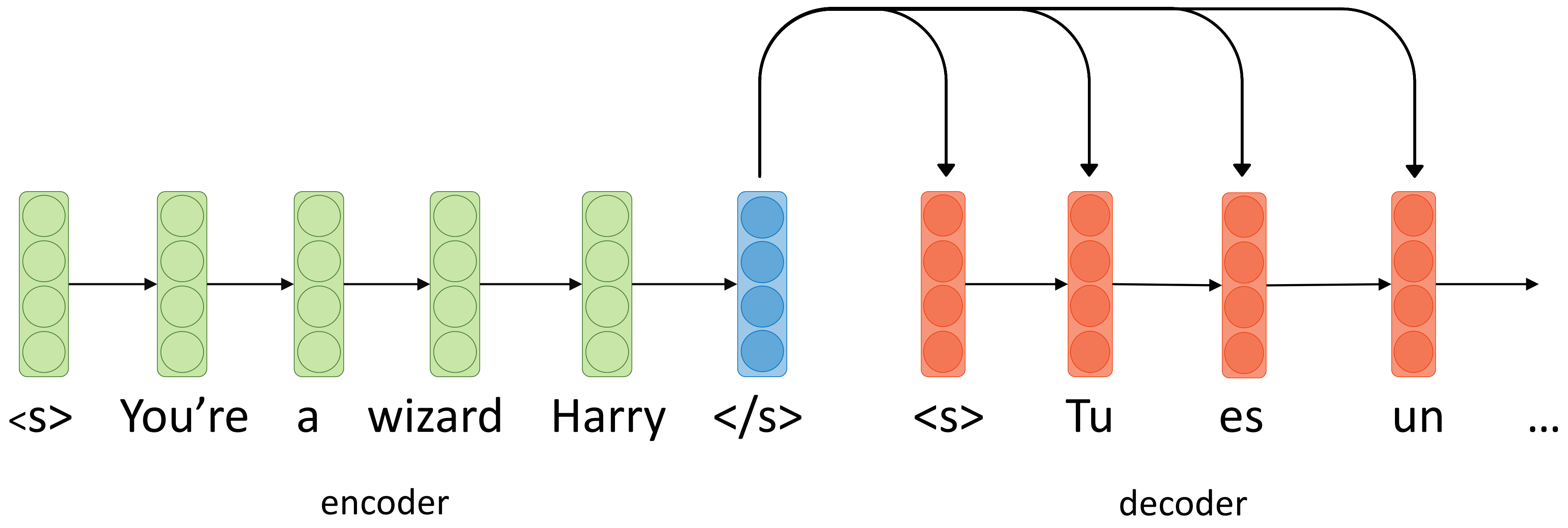

X
Y
Z
</s>
A
B
C
D
<s>
X
Y
Z


Tu
es
un
sorcier
You're
a
wizard
Harry
<s>
Tu
es
un


<s> You’re a wizard Harry </s>
<s> Tu es un …
encoder
decoder


context vector
<s>
You’re
a
wizard
Harry
</s>
<s>
Tu
es
un
...
encoder
decoder

Cho et al., (2014): connect to every state in decoder



Attention

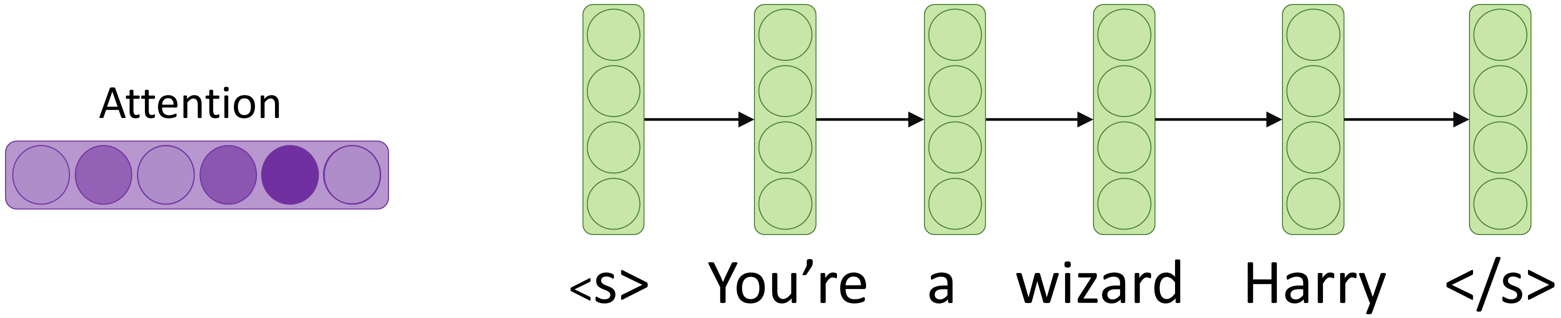
- Soft version of alignment
- Represents the importance of each word in input for predicting a word in output
- We’ll talk about how much the network “attends” to each word.
- First used in MT, improves BLEU score by 10 pts

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. (2014) **Neural machine translation by jointly learning to align and translate**.

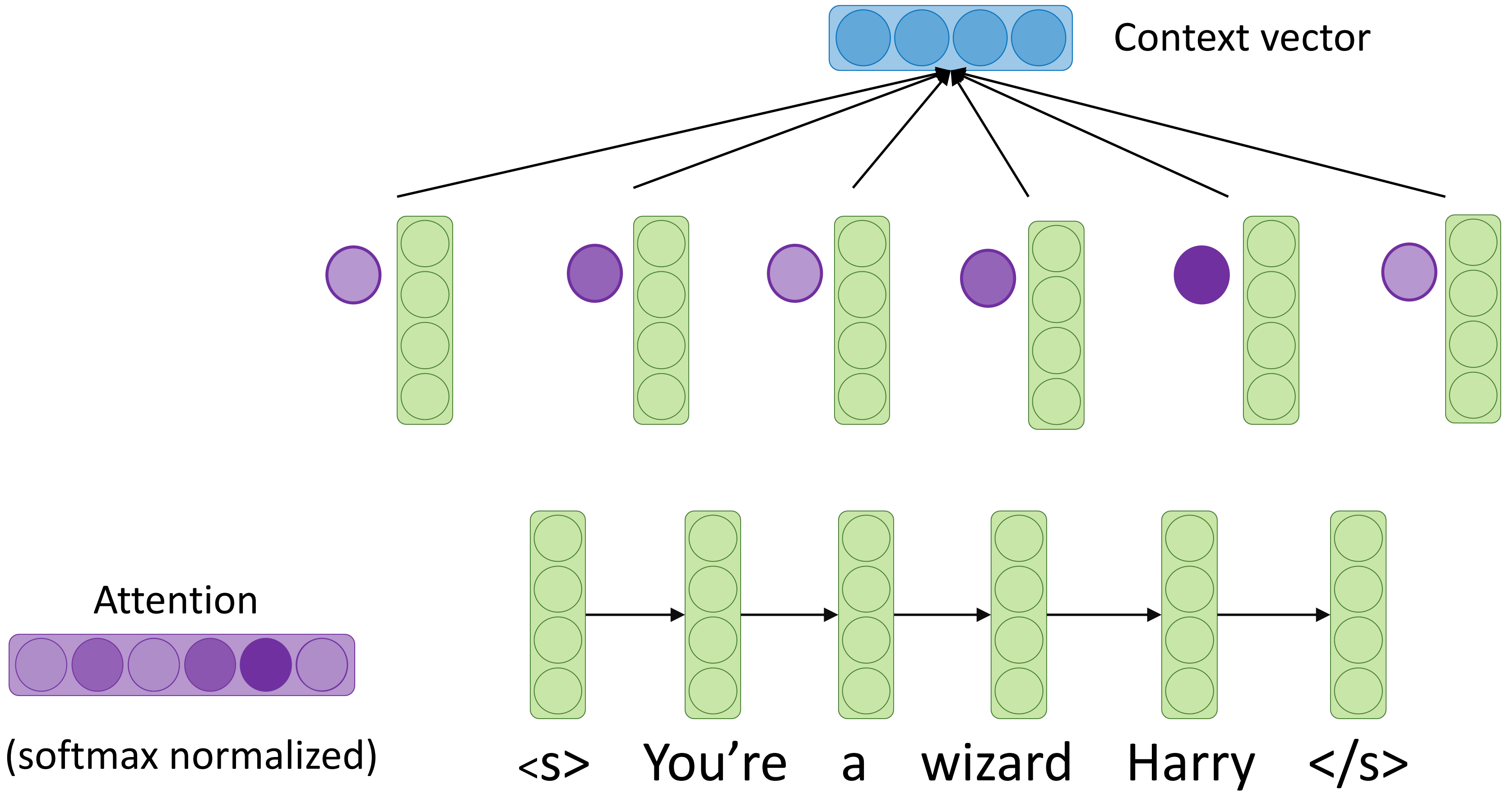
Activity

Question									Answer		
What	is	the	preferred	weapon	of	the	Jedi	?	a	light	saber
PRON	AUX	DET	ADJ	NOUN	ADP	DET	PROPN		1	2	3

Attention

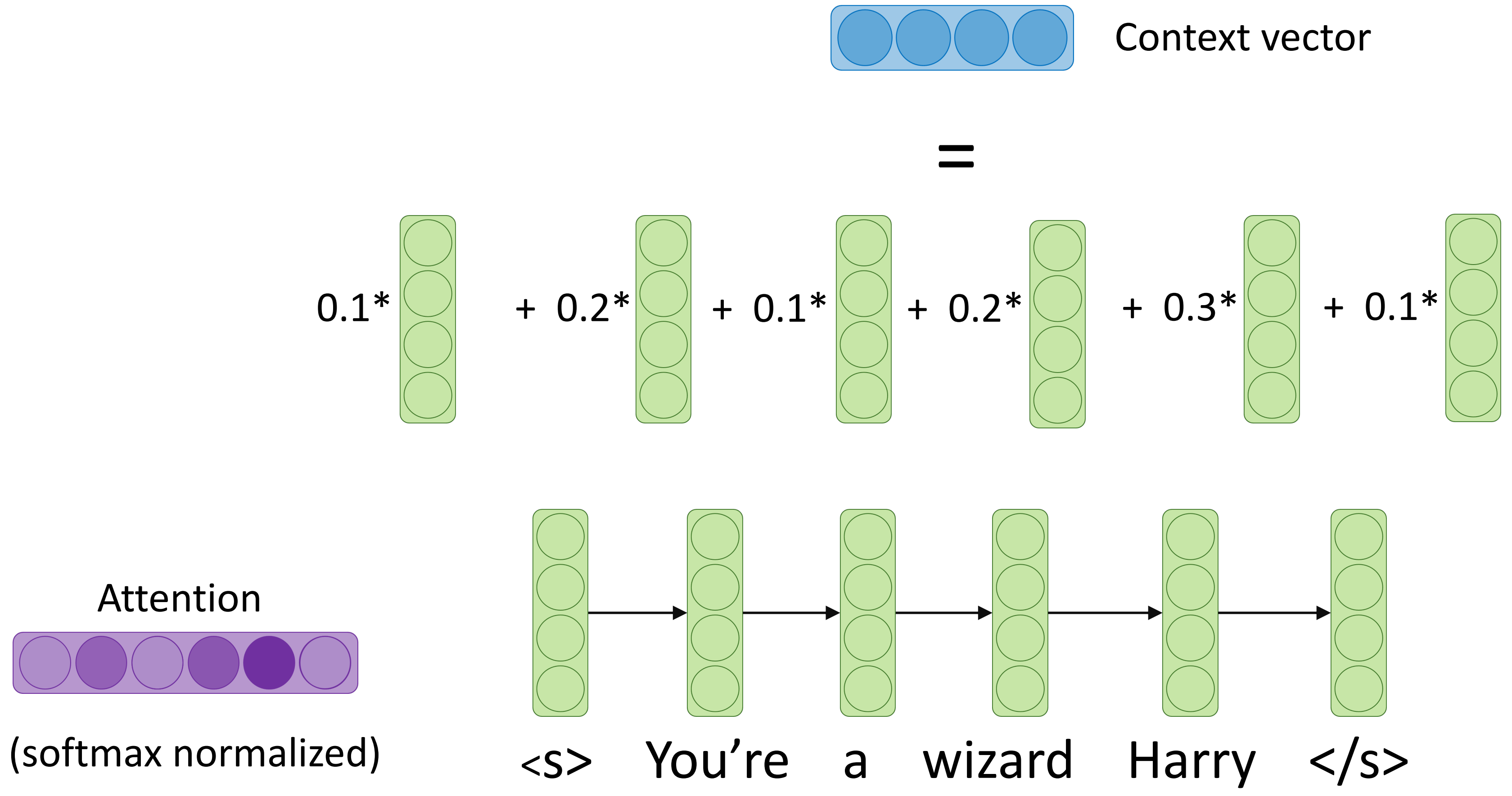


Attention



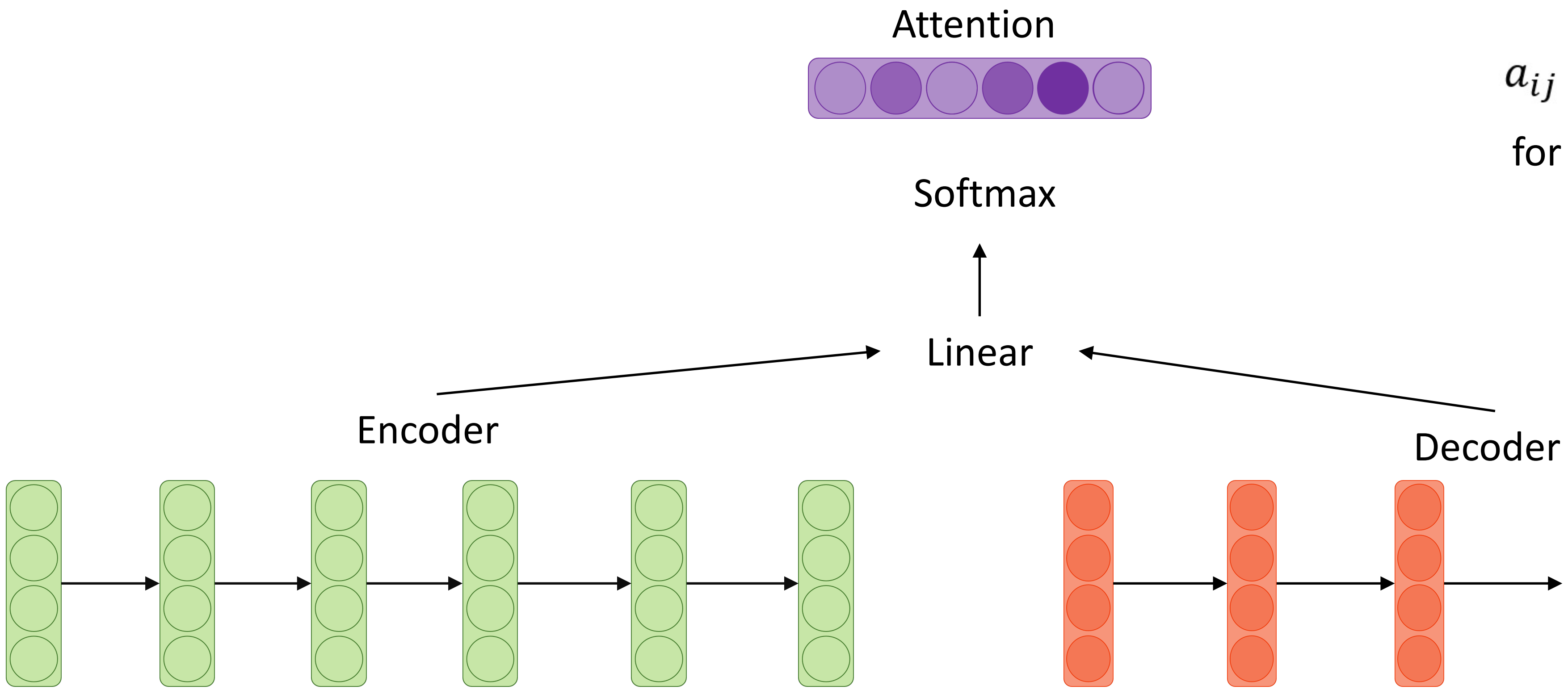
$$c_i = \sum_{j=1} \alpha_{ij} h_j$$

Attention



$$c_i = \sum_{j=1} \alpha_{ij} h_j$$

Attention

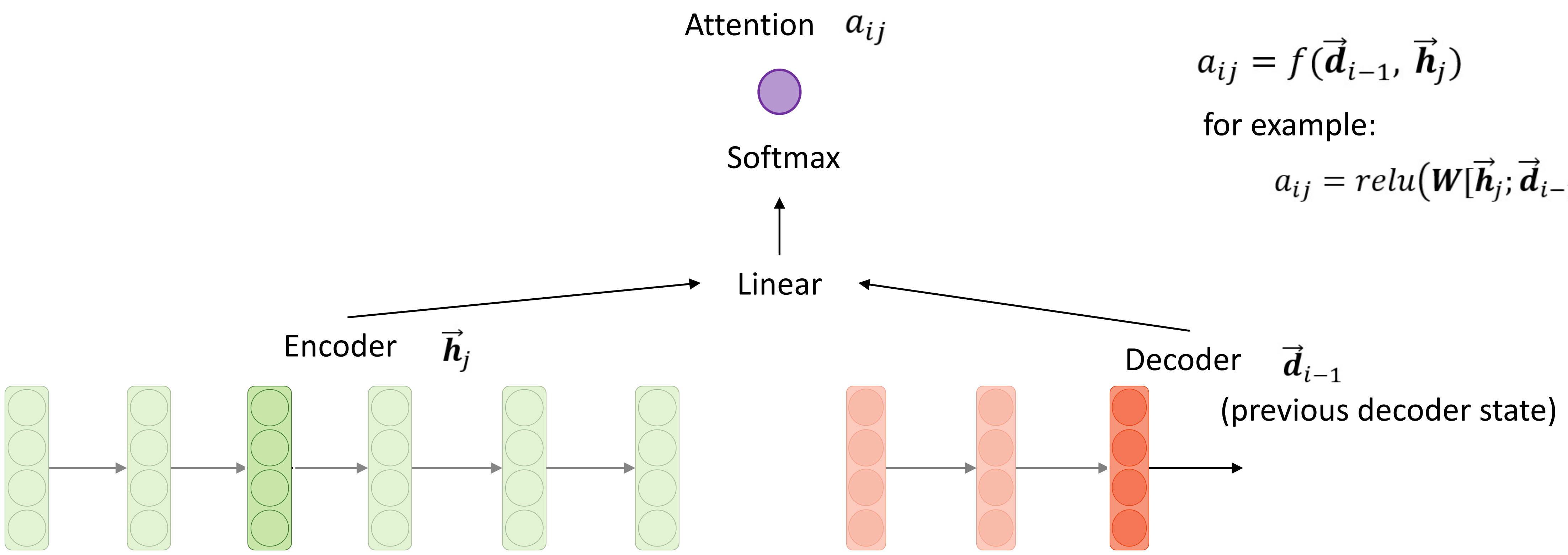


$$a_{ij} = f(\vec{\boldsymbol{d}}_{i-1}, \vec{\boldsymbol{h}}_j)$$

for example:

$$a_{ij} = relu\left(\boldsymbol{W}[\vec{\boldsymbol{h}}_j; \vec{\boldsymbol{d}}_{i-1}]\right)$$

Attention

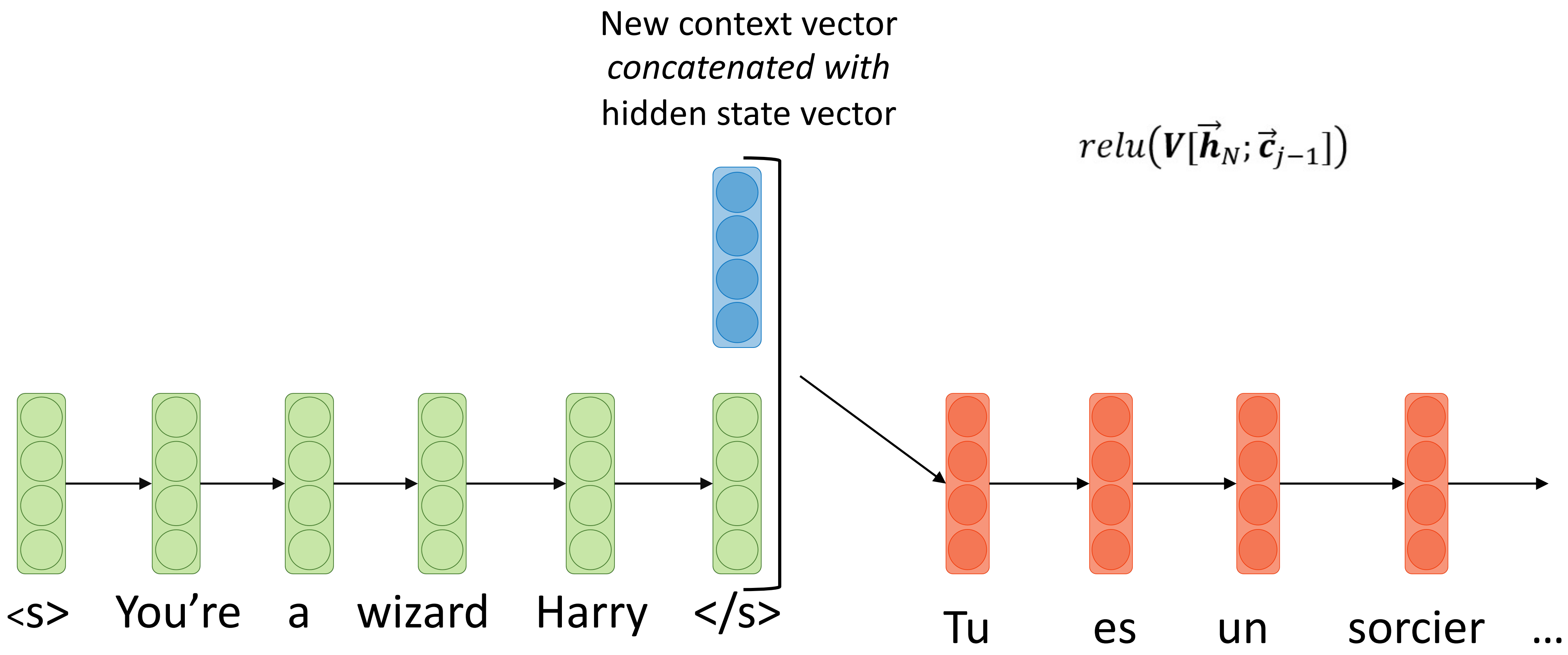


$$a_{ij} = f(\vec{\boldsymbol{d}}_{i-1}, \vec{\boldsymbol{h}}_j)$$

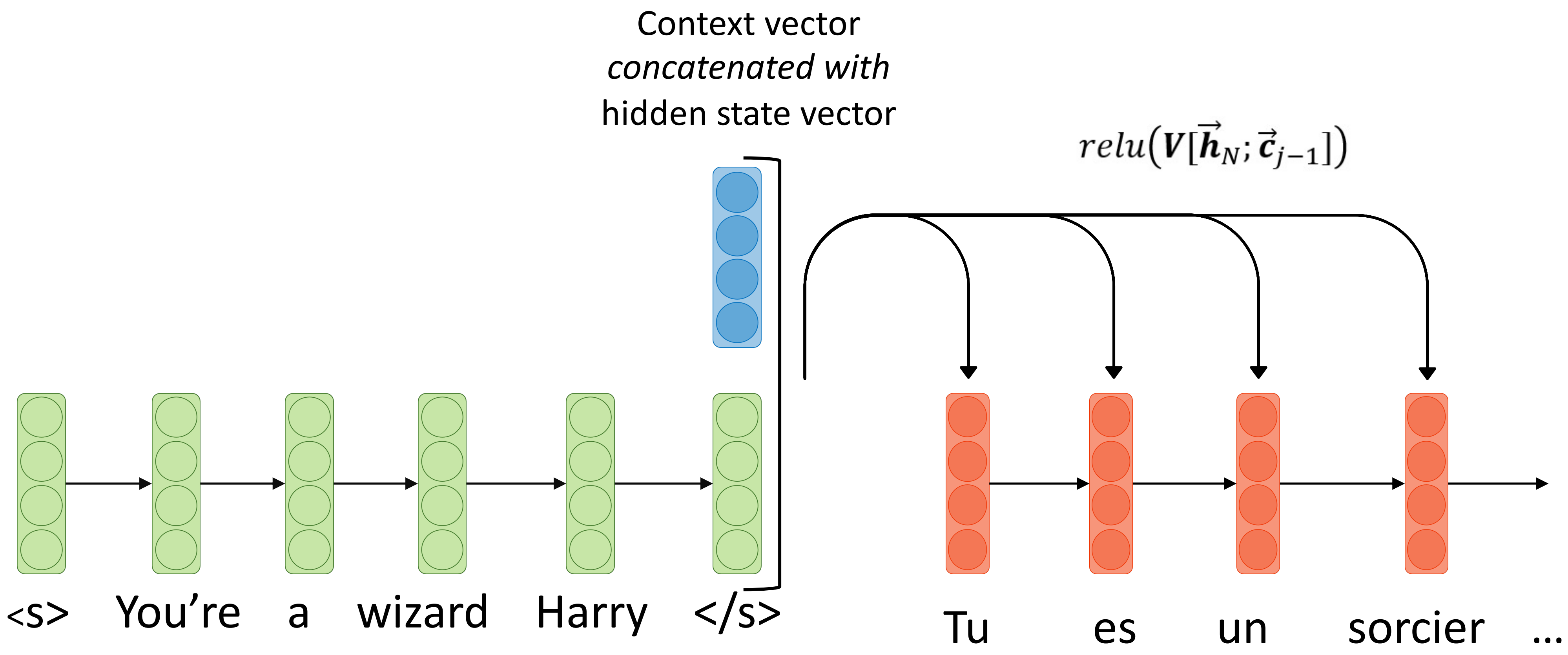
for example:

$$a_{ij} = relu(\boldsymbol{W}[\vec{\boldsymbol{h}}_j; \vec{\boldsymbol{d}}_{i-1}])$$

Attention


New context vector
concatenated with
hidden state vector
$relu(\boldsymbol{V}[\vec{\boldsymbol{h}}_N; \vec{\boldsymbol{c}}_{j-1}])$
<s>
You're
a
wizard
Harry
</s>
Tu
es
un
sorcier
…

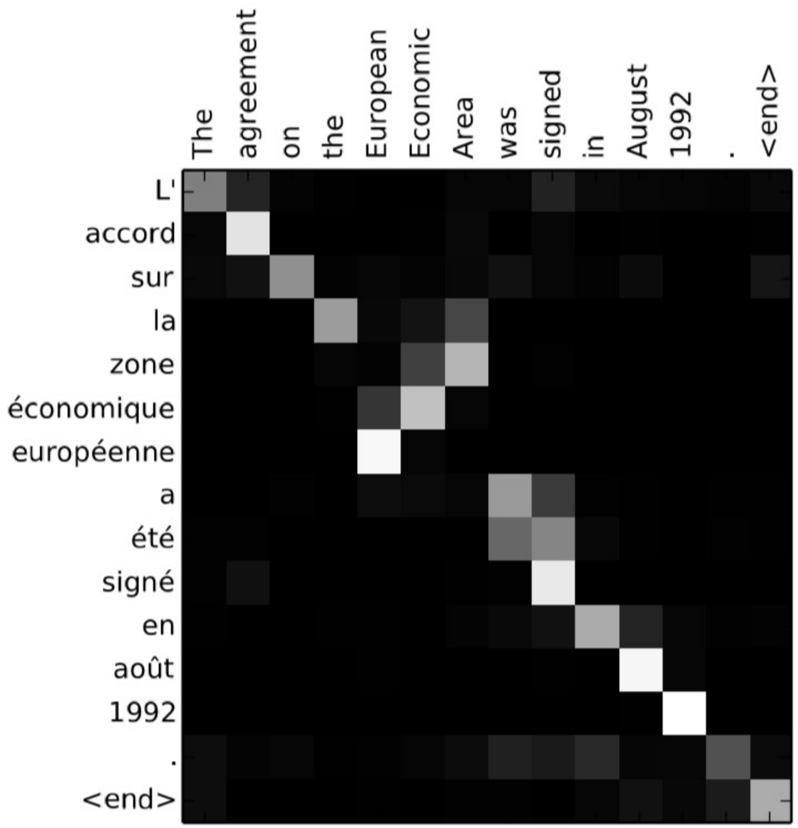
Attention


Context vector
concatenated with
hidden state vector
$relu(\boldsymbol{V}[\vec{\boldsymbol{h}}_N; \vec{\boldsymbol{c}}_{j-1}])$
<s>
You're
a
wizard
Harry
</s>
Tu
es
un
sorcier
...

Attention

Every attention value depends on one word in the source and one in the target.

Attention matrix tells us how “important” a source word is for each target word (much like alignment).



Transformers

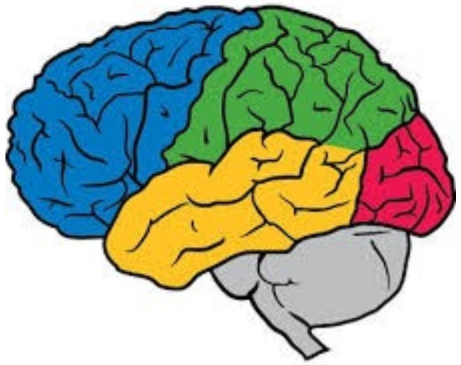
Attention is All You Need (Vaswani et al., 2017)

Debut of the **Transformer** architecture.

The same model used in:

- BERT (Devlin, et al. 2018)
- LISA (Strubell, et al. 2018)
- RoBERTa (Liu et al., 2019)
- and others…

Motto paraphrased: *No more RNNs, CNNs, just use Attention!*



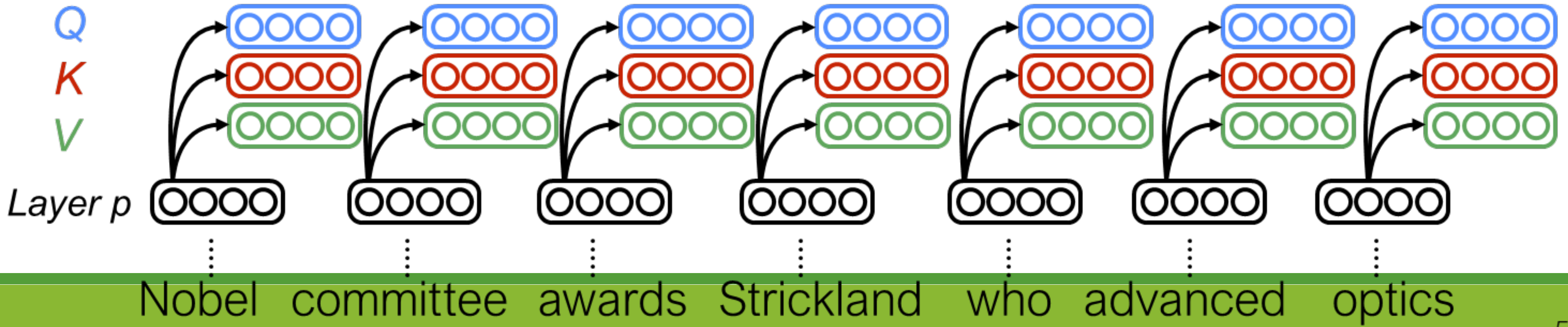
Transformer: Self-Attention

$$\left.\begin{array}{ll}\text{query} & {\color{blue}Q} \\ \text{key} & {\color{red}K}\end{array}\right\} \text{used to calculate attention}$$

$$\begin{array}{ll}\text{value} & {\color{green}V}\end{array} \text{ — base representation of word}$$

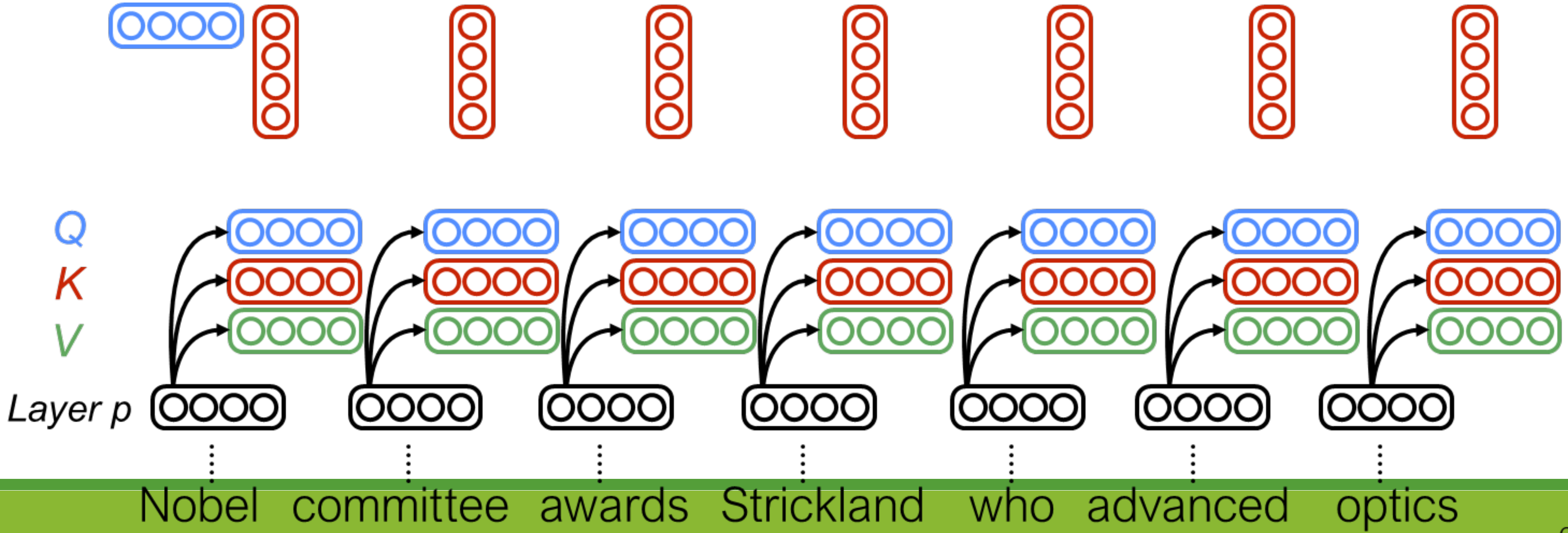
Transformer: Self-Attention

Slide from Strubell, et al. (2018).



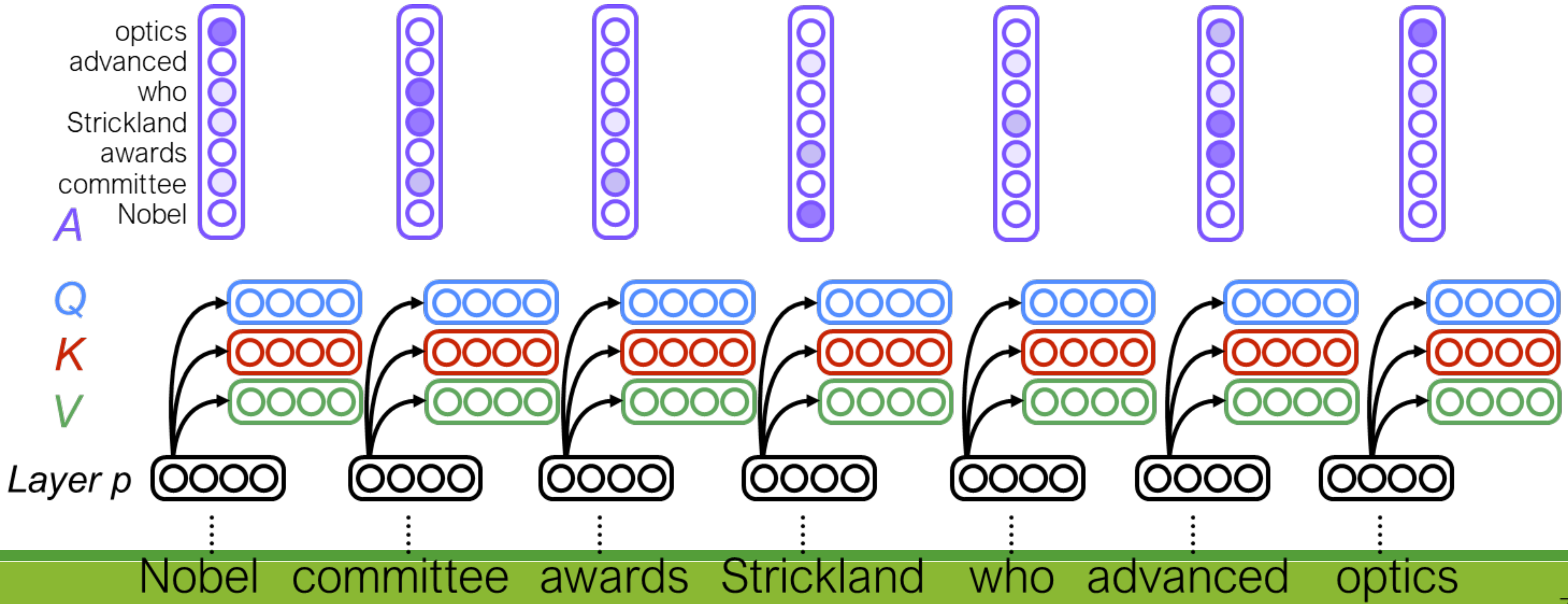
Transformer: Self-Attention

Slide from Strubell, et al. (2018).



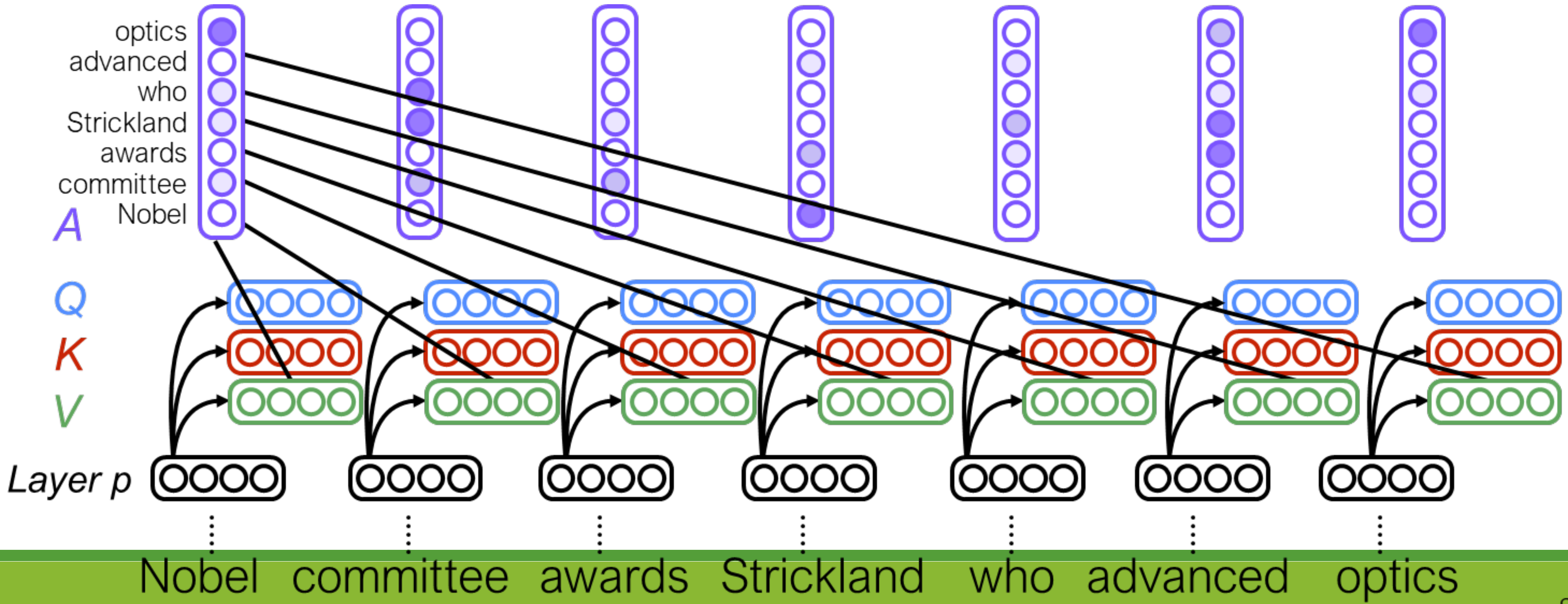
Transformer: Self-Attention

Slide from Strubell, et al. (2018).



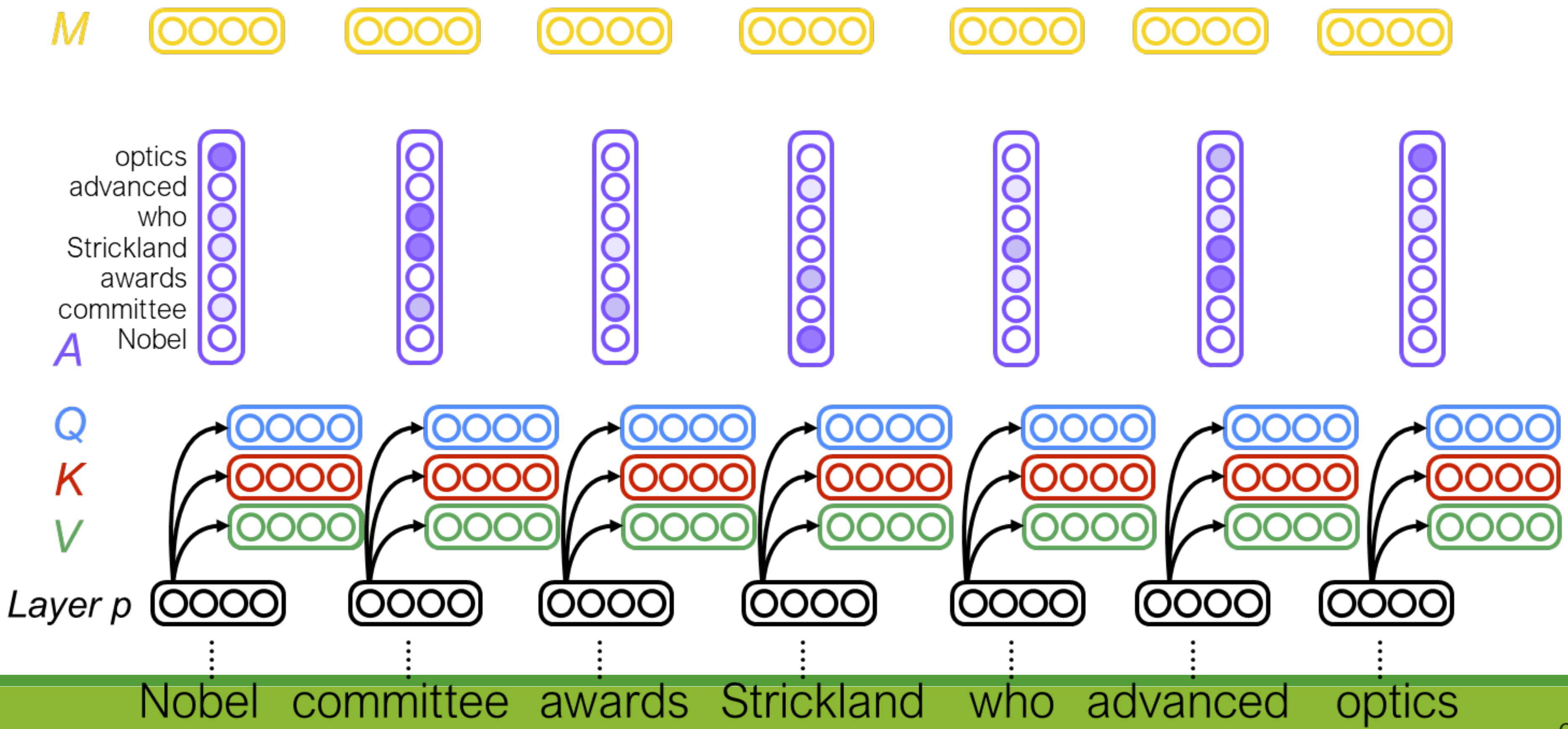
Transformer: Self-Attention

Slide from Strubell, et al. (2018).



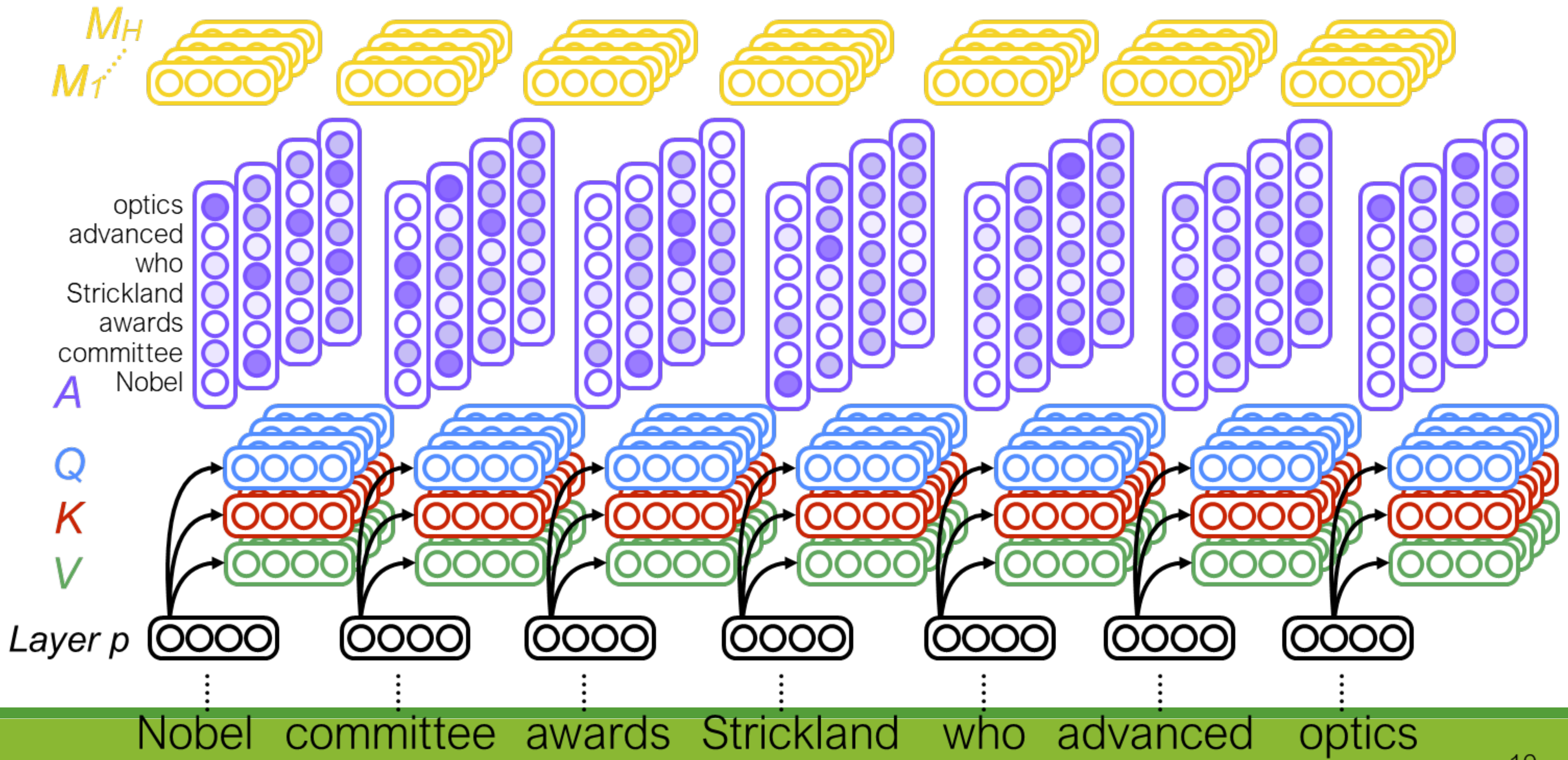
Transformer: Self-Attention

Slide from Strubell, et al. (2018).



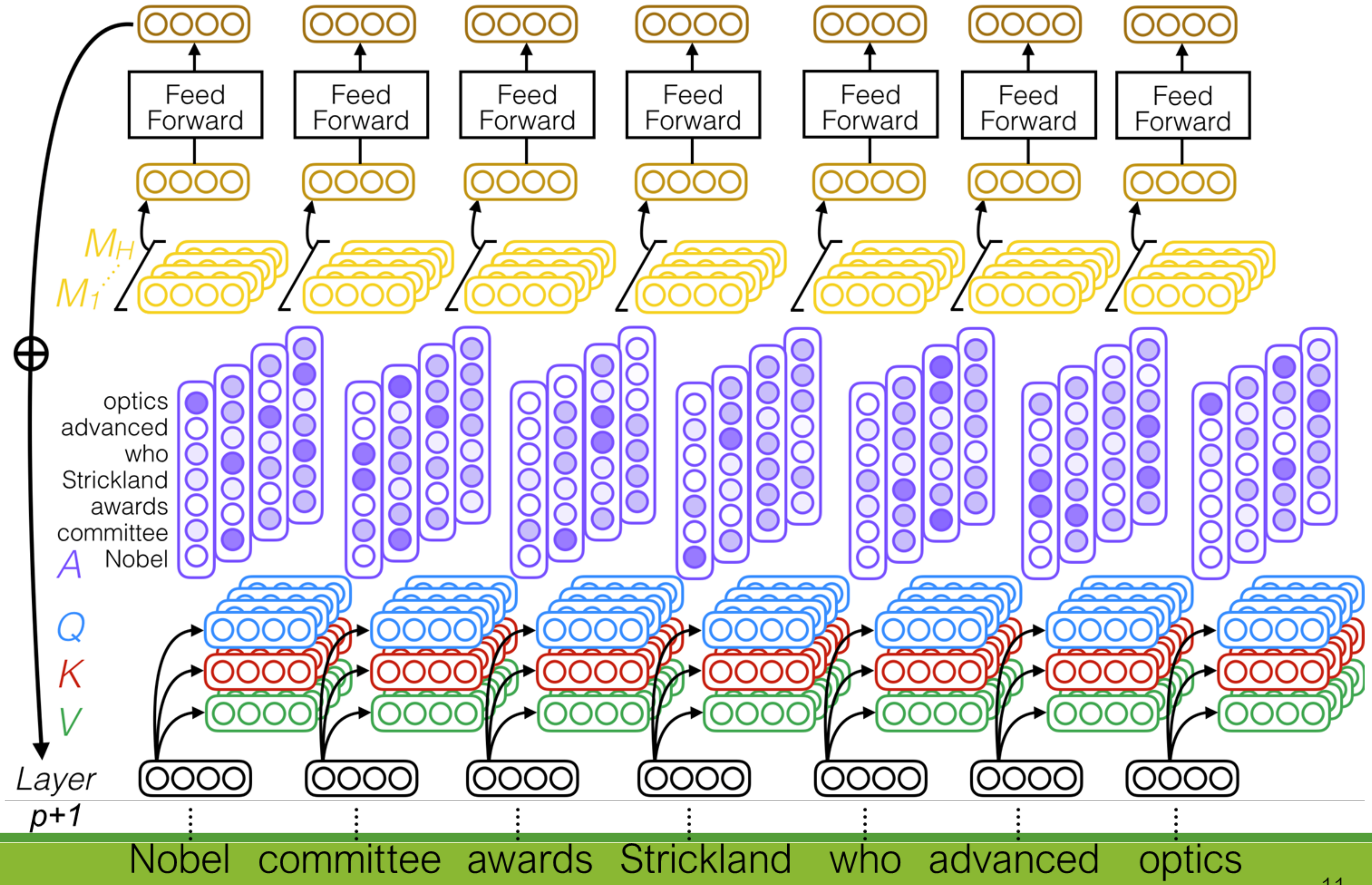
Transformer: (Multi-head) Self-Attention

Slide from Strubell, et al. (2018).



Transformer: (Multi-head) Self-Attention

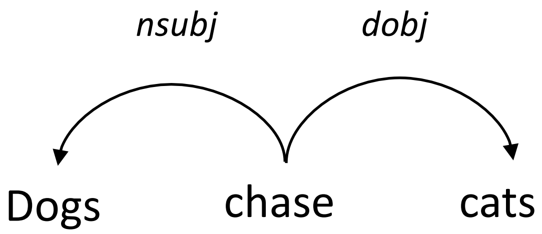
Slide from Strubell, et al. (2018).

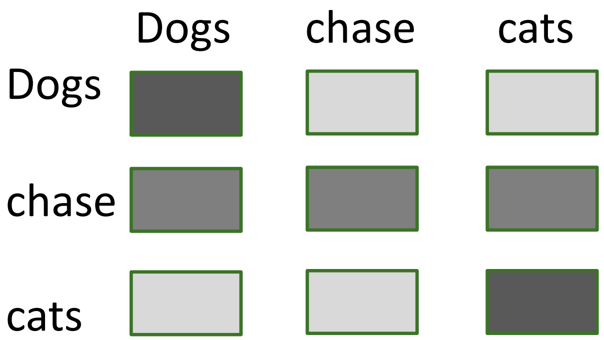


Linguistically Motivated

Strengths

- Captures long-distance dependencies!
- Intuitively: approximates *weighted unlabeled dependencies*

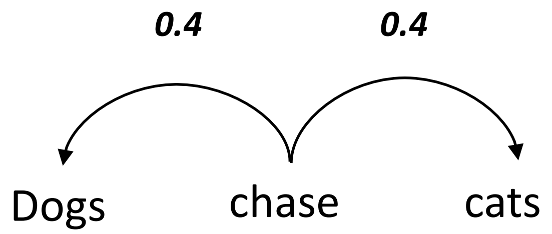


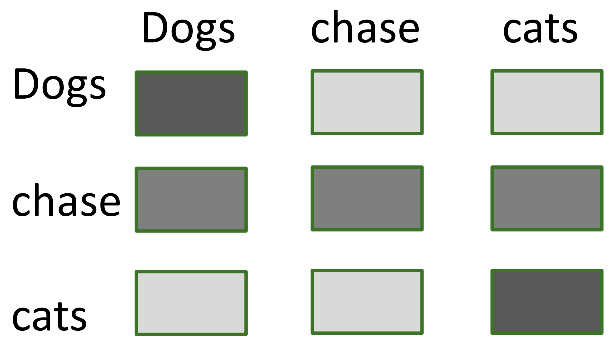


Linguistically Motivated

Strengths

- Captures long-distance dependencies!
- Intuitively: approximates *weighted unlabeled dependencies*

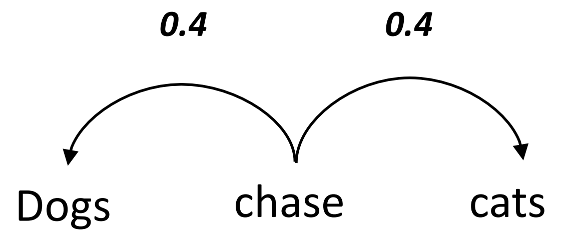




Linguistically Motivated

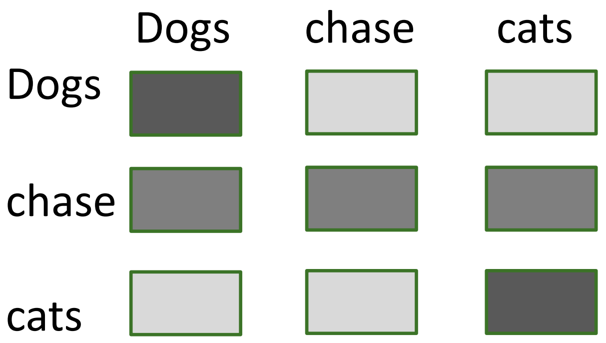
Strengths

- Captures long-distance dependencies!
- Intuitively: approximates *weighted unlabeled dependencies*



Weaknesses (addressed in next slides)

- Weak model of word order
- One layer can't distinguish dependencies
- No locality bias



Positional Encoding

How to turn this into a sequence modal:

Add “positional encoding” as extra input.
Weak representation of position.

Possible Improvement: Unlike RNN and CNN,
No locality bias.

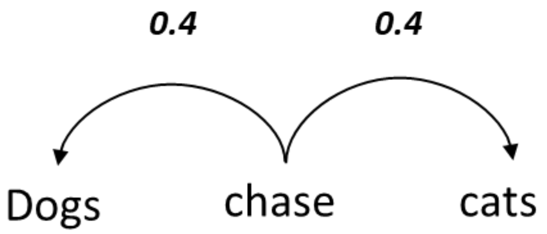
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

(pos is position, i is dimension)

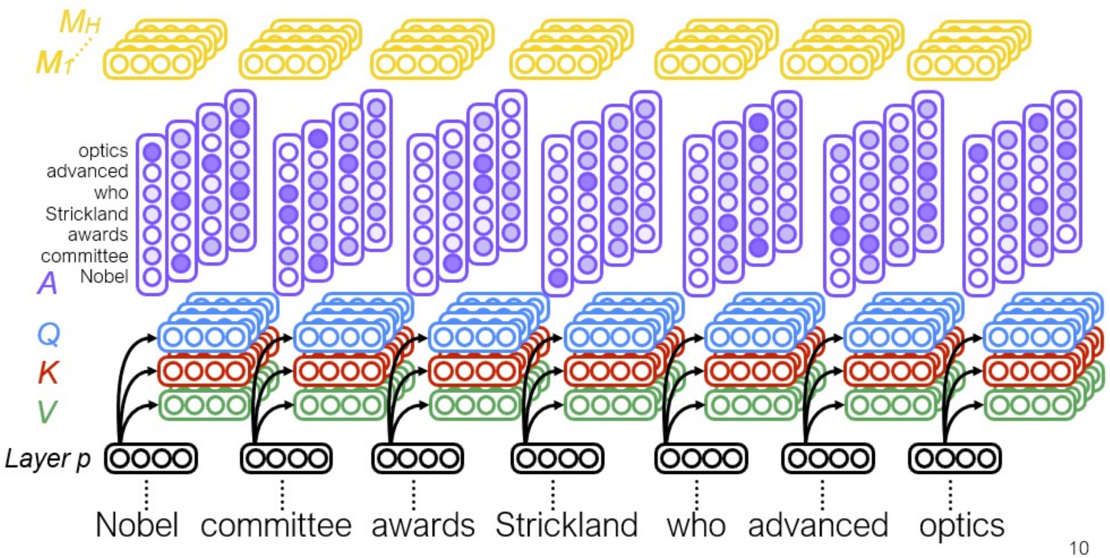
Multi-Head Attention

How to distinguish dependencies:

One attention layer can't distinguish two dependencies (subject vs. object).



Use multiple attention layers, hopefully one represents subject, one object, etc.



Transformer Architecture

- Encode-Decoder with Transformers instead of RNNs
- Large improvement over LSTM encoder-decoder. Why?

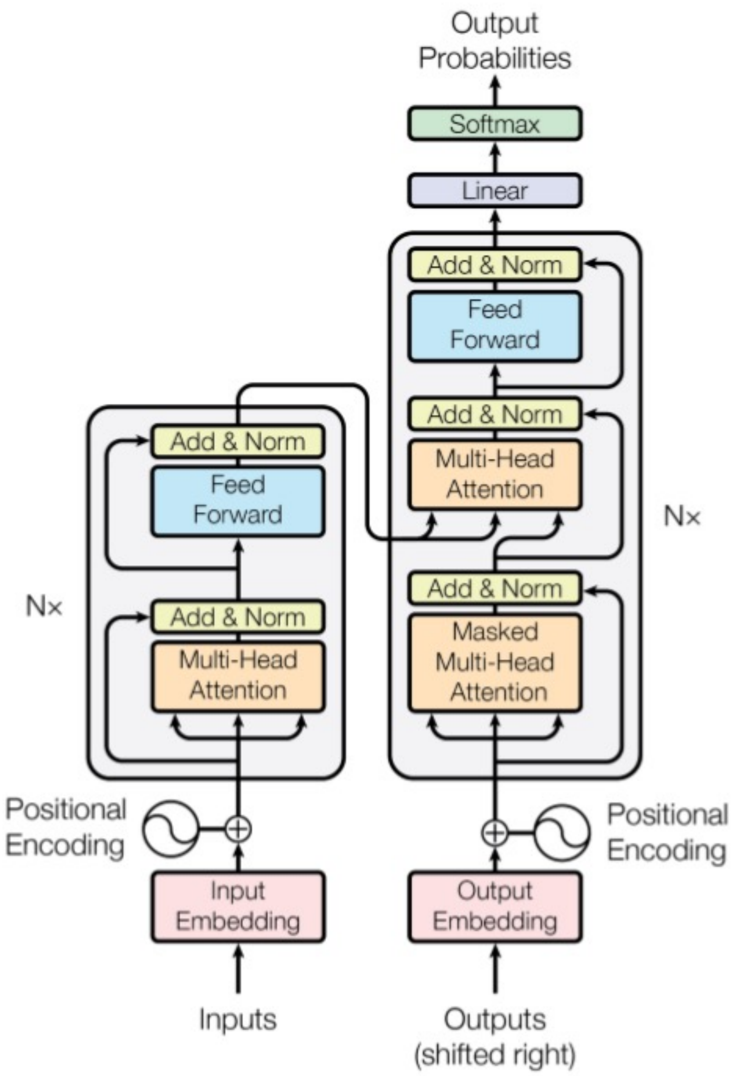


Figure 1: The Transformer - model architecture.

Transformer Architecture

- Encode-Decoder with Transformers instead of RNNs
- Large improvement over LSTM encoder-decoder. Why?
  - long-distance relations
  - better representation of syntax
  - faster to train (when using TPUs)

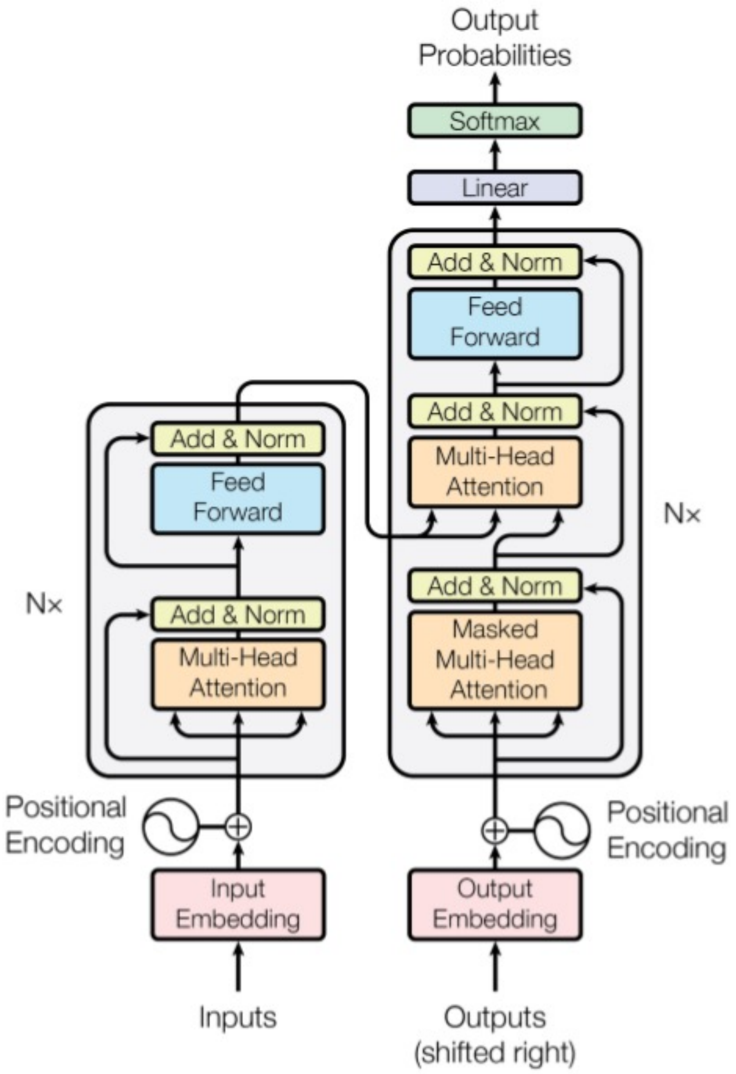


Figure 1: The Transformer - model architecture.

Replicate, Extend, etc.

Tensorflow

https://github.com/tensorflow/tensor2tensor

Pytorch

https://github.com/jadore801120/attention-is-all-you-need-pytorch

Annotated Code

http://nlp.seas.harvard.edu/2018/04/03/attention.html

Illustrated Explanation

http://jalammar.github.io/illustrated-transformer/

Transfer Learning: ELMo & BERT

Why Transfer Learning

- “Free” increases in accuracy
- For some tasks, data is sparse or expensive (AMR, low resource languages, etc.)
- May capture information that is useful but not present in labelled data.
- Possibly closer to genuine linguistic representations.

Review: Word Embeddings

GloVe is a collection of pretrained (static) word embeddings that can be plugged into your models.

Approximate semantic features: King - Man + Woman = Queen

Trained on millions of sentences

Can be “tuned”: your model can adjust GloVe features to be more useful for your task.

Pennington, J., Socher, R., & Manning, C. (2014). **Glove: Global vectors for word representation**. In *Proceedings of the 2014 conference on EMNLP.*

ELMo: Deep contextualized word representations (Peters et al., 2018)

Embeddings from **L**anguage **Mo**dels (ELMo)

Word vector representations that is a function of input sentence.

Based on BiLSTM language model.

Motivation

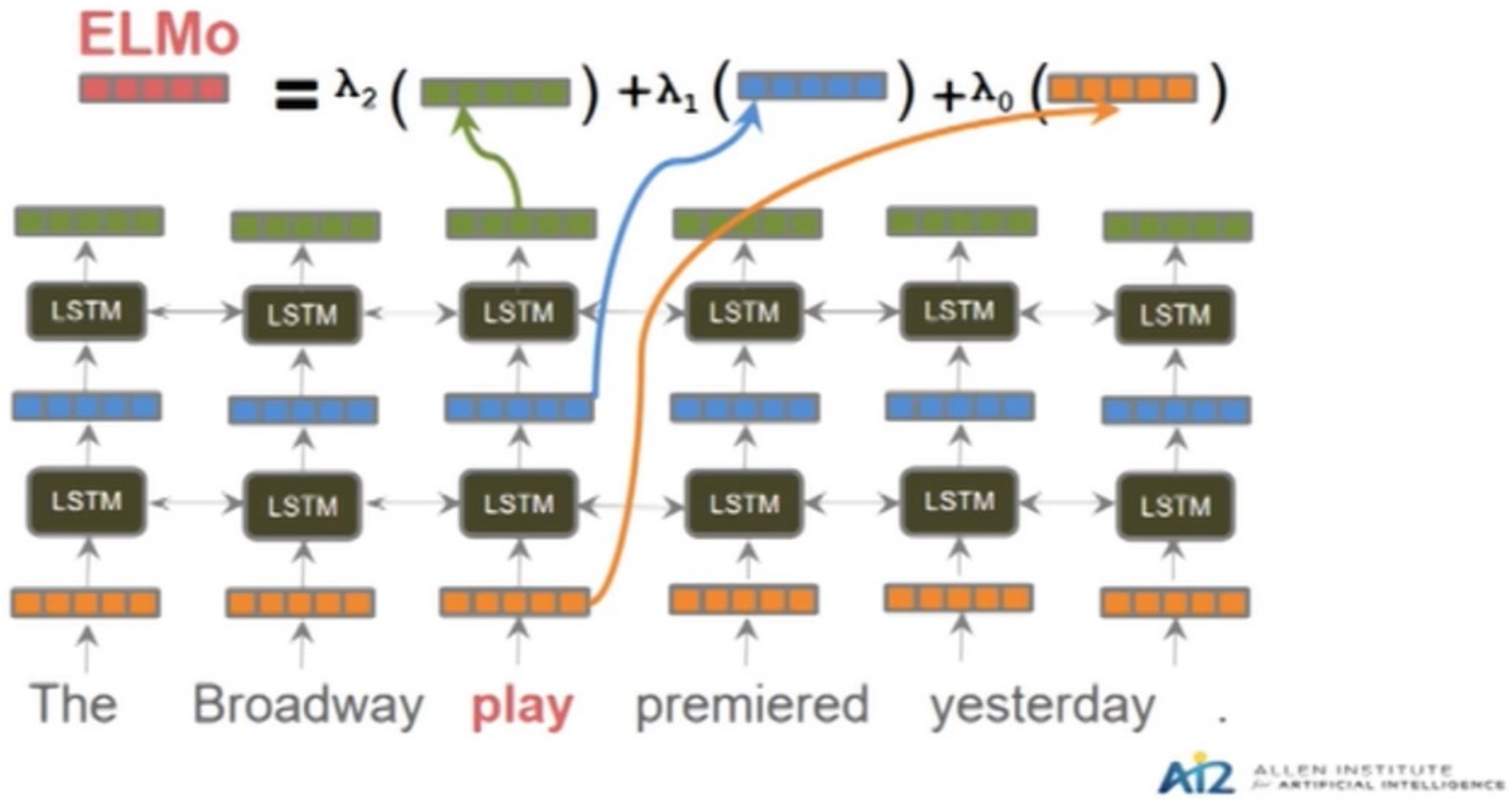
Previous embedding models fail to address *polysemy* or *orthographic variation* (morphology).

Idea:
Build pre-trained word embeddings that are a function of the input sentence.

polysemy	The Broadway play premiers tomorrow. Let’s play outside.
morphology	play, plays, playing, multiplayer, Play

slide from
(Peters et al., 2018)

Embeddings from Language Models



Replicate, Extend, etc.

https://allennlp.org/elmo

Tensorflow

https://github.com/allenai/bilm-tf

Pytorch

https://github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2018)

Bidirectional **E**ncoder **R**epresentations from **T**ransformers
Transfer learning using transformers and new prediction tasks.

Masked LM

slides from Devlin et al. (2018)

- **Solution**: Mask out *k*% of the input words, and then predict the masked words
  - We always use k = 15%

```
                       store            gallon
                         ↑                 ↑
the man went to the [MASK] to buy a [MASK] of milk
```

- Too little masking: Too expensive to train
- Too much masking: Not enough context

Masked LM

slides from Devlin et al. (2018)

- Problem: Mask token never seen at fine-tuning
- Solution: 15% of the words to predict, but don't replace with `[MASK]` 100% of the time. Instead:
- 80% of the time, replace with `[MASK]`

  `went to the store → went to the [MASK]`
- 10% of the time, replace random word

  `went to the store → went to the running`
- 10% of the time, keep same

  `went to the store → went to the store`

Next Sentence Prediction

slides from Devlin et al. (2018)

- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

Experiments

- GLUE: **Textual Inference** (MNLI, RTE, WNLI), **Question Similarity** (QQP), **Question Answering** (QNLI), **Sentiment Analysis** (SST-2), **Grammaticality** (CoLa), **Semantic Similarity** (STS-B, MRPC)
- SQuAD (Question Answering)
- Named Entity Recognition
- SWAG (Adverserial Sentence Prediction)

Experiments: GLUE

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\mathrm{BERT}_{\mathrm{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\mathrm{BERT}_{\mathrm{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); $\mathrm{BERT}_{\mathrm{BASE}}$ = (L=12, H=768, A=12); $\mathrm{BERT}_{\mathrm{LARGE}}$ = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from https://gluebenchmark.com/leaderboard and https://blog.openai.com/language-unsupervised/.

Demo

AllenNLP demo of BERT

https://demo.allennlp.org/masked-lm

Replicate, Extend, etc.

Includes 104 languages

Tensorflow

https://github.com/google-research/bert

Pytorch

https://github.com/huggingface/pytorch-pretrained-BERT

Recent Transfer Learning Projects

RoBERTa (Liu et al., 2019)

https://github.com/pytorch/fairseq

XLNet (Yang et al., 2019)

https://github.com/zihangdai/xlnet