More seq2seq
Inverse Problems
LM-Pretraining

Yoav Goldberg

Last time

- Attention
- Seq2Seq + Attention
- The attention abstraction
- Transformers
    - Self-attention
    - Multi-head attention

Transformers: Problems

- No clear computational model.
- Can be parallelized on the GPU, but computation is still expensive.
- There is an n^2 memory dependence on sequence length --> this severely limits modeled sequence length.

Transformers: current research

- What is the computational model behind a transformer?

- Can we make transformers cheaper by removing the n^2 dependence on length? (e.g, "Longformer")

- Can we remove the dependence on the attention operation? can we replace attention with something cheaper?

ViT: Visual Transformer

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Alexey Dosovitskiy[*,†], Lucas Beyer[*], Alexander Kolesnikov[*], Dirk Weissenborn[*], Xiaohua Zhai[*], Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby[*,†]

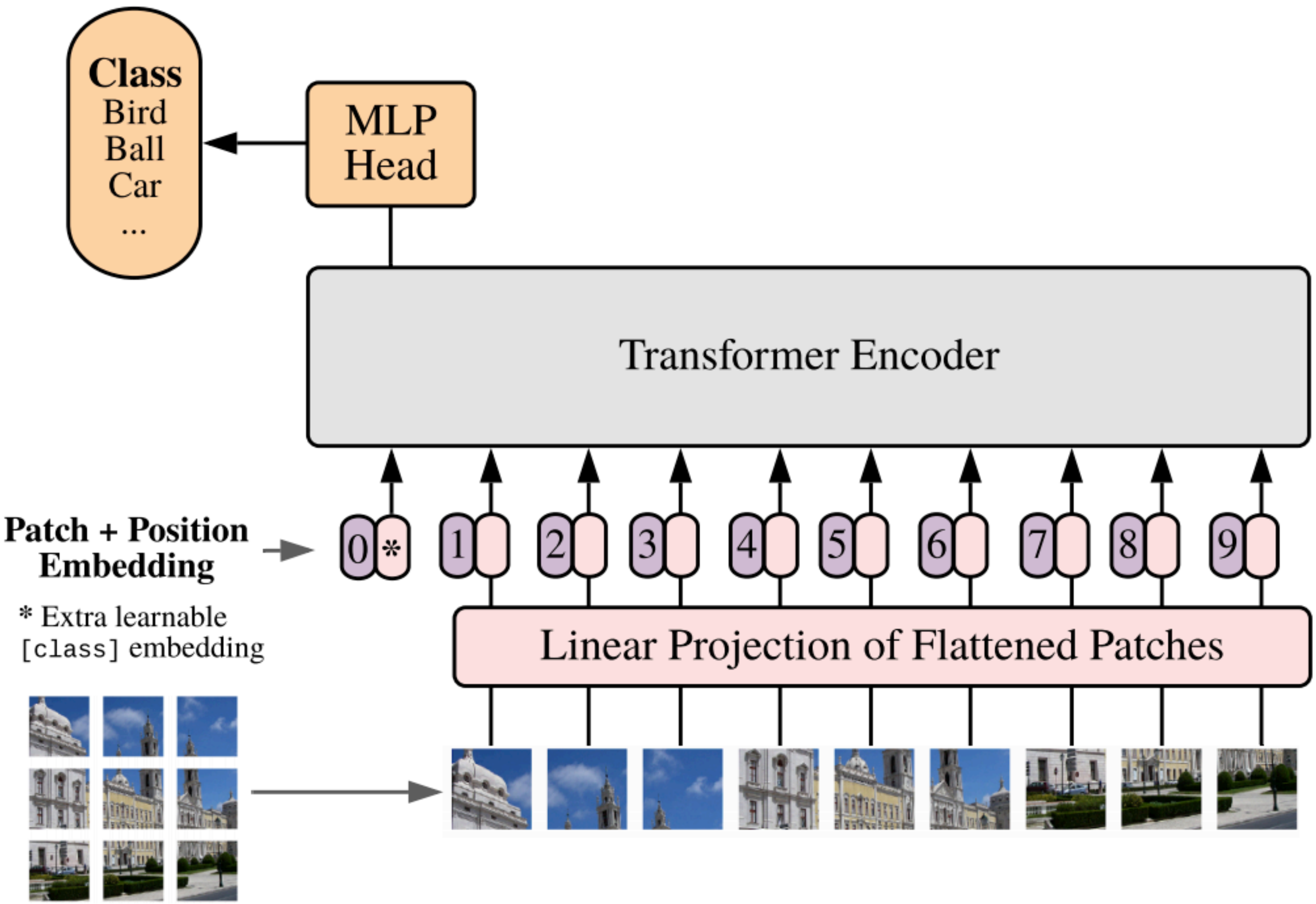
[*]equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

"with enough data and rich models, things tend to work"

Vision Transformer (ViT)



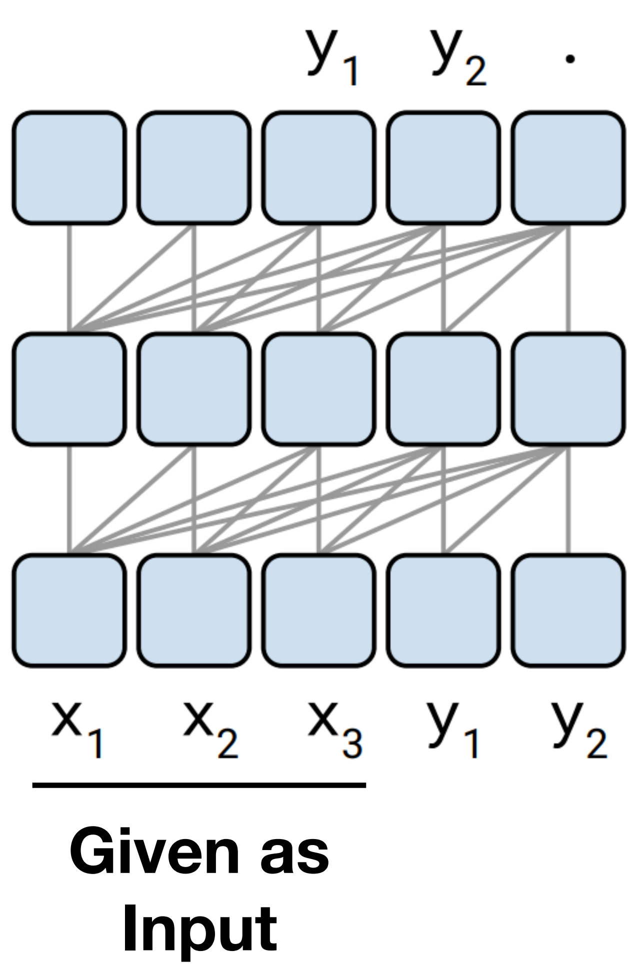
back to seq2seq / enc-dec

x1 x2 x3 --> y1 y2 .

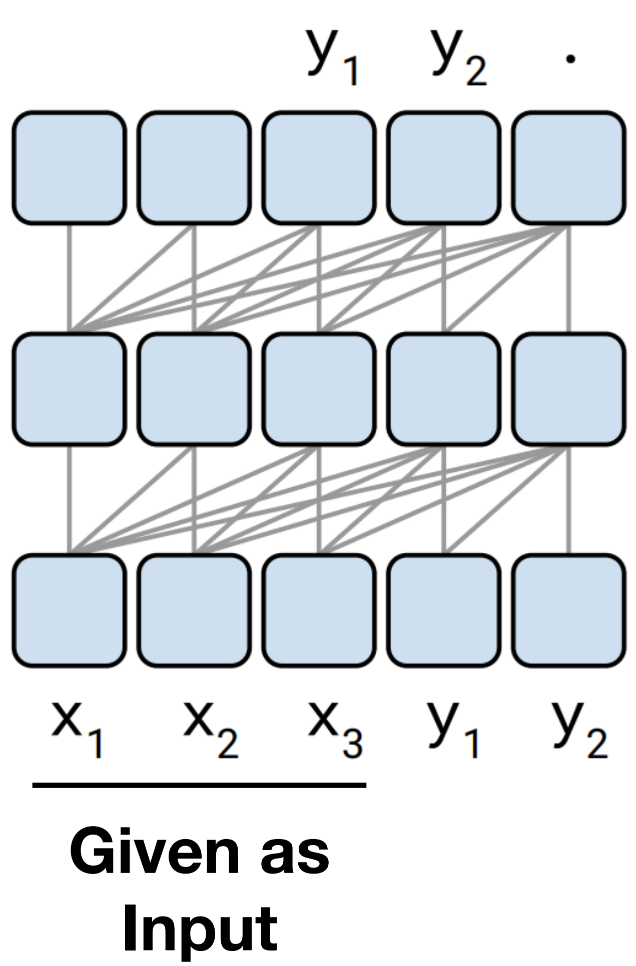
. \

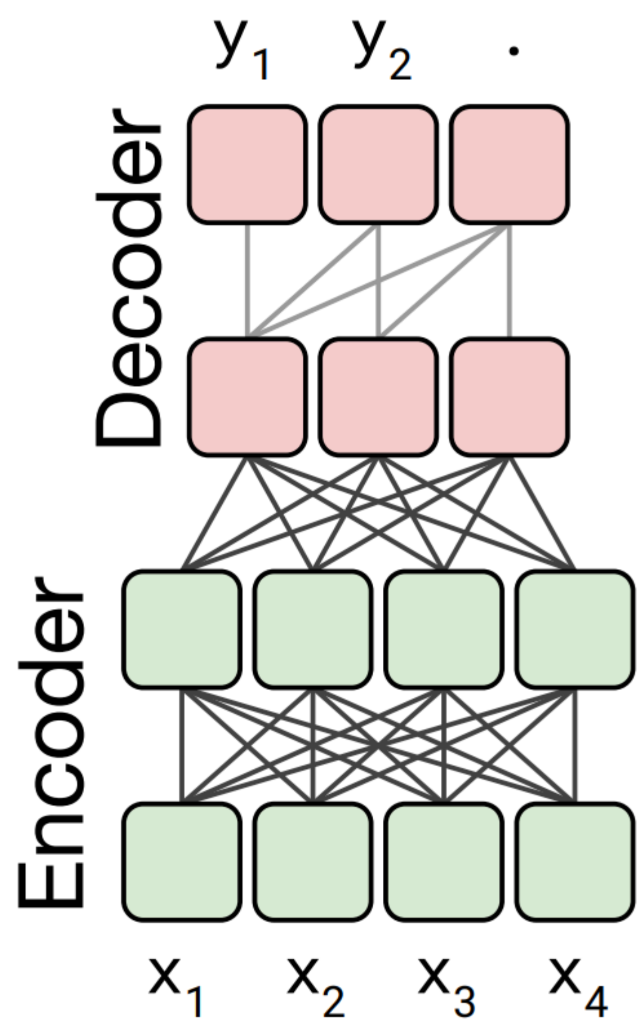
(slide by Graham Neubig)

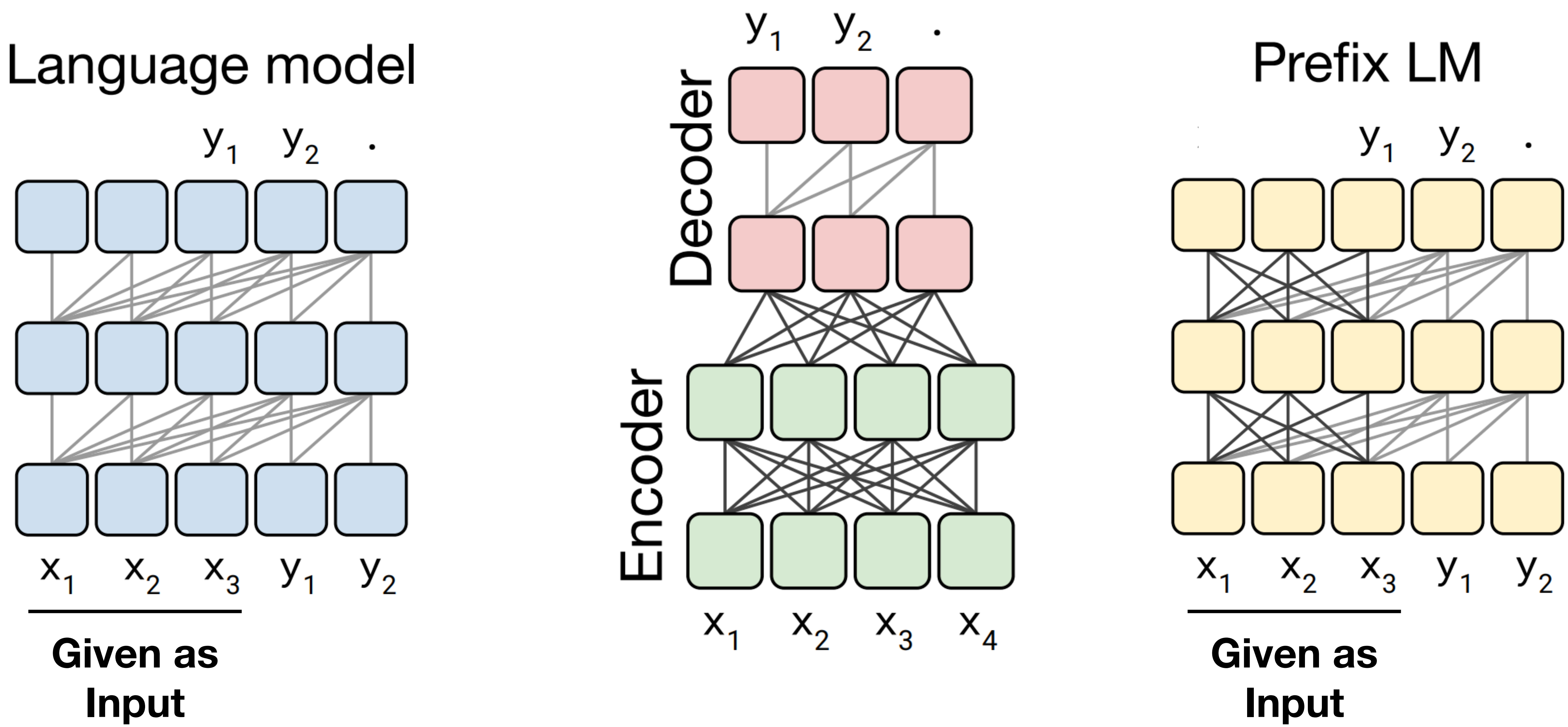
Language model



Language model

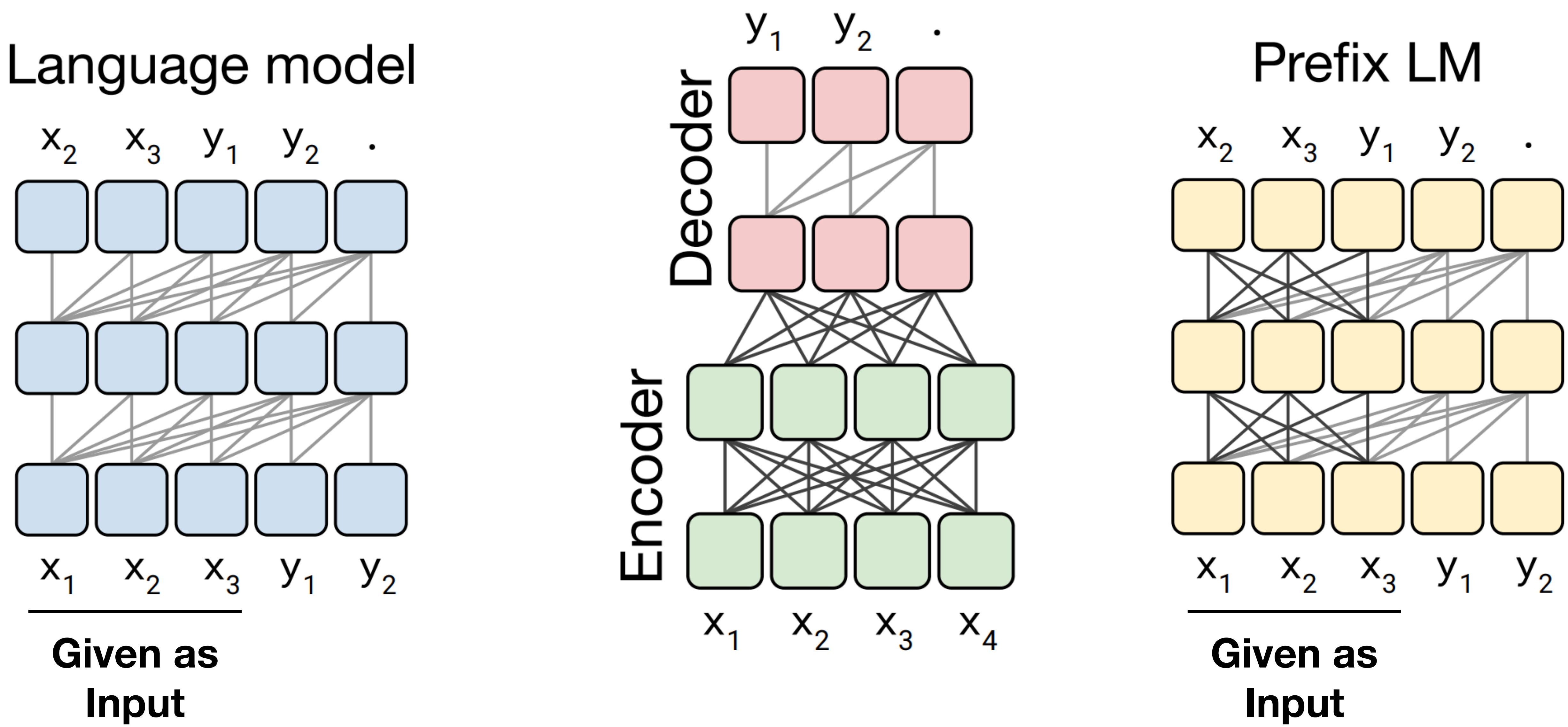






Language model
y1 y2 .
x1 x2 x3 y1 y2
Given as Input
y1 y2 .
Decoder
Encoder
x1 x2 x3 x4
Prefix LM
y1 y2 .
x1 x2 x3 y1 y2
Given as Input

when training LM / Prefix-LM, can model also x2, x3



Masked Attention

How do we decode with a Transformer model?

How do we implement it?

(efficiently)

(at training time)

Masked Attention

- We want to perform training in as few operations as possible using big matrix multiplies
- We can do so by “masking” the results for the output

(slide by Graham Neubig)

Applications of seq2seq

- previously: translation, summarization, email response, dialog...

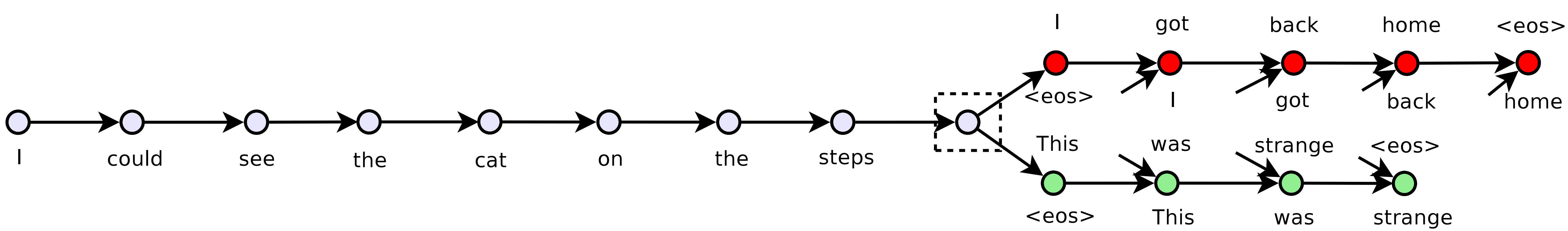
Skip-Thought Vectors

Ryan Kiros [1], Yukun Zhu [1], Ruslan Salakhutdinov [1,2], Richard S. Zemel [1,2]
Antonio Torralba [3], Raquel Urtasun [1], Sanja Fidler [1]
University of Toronto [1]
Canadian Institute for Advanced Research [2]
Massachusetts Institute of Technology [3]

- Really cheesy name.
- Really cool idea.

- Generalize distributional similarity to sentences.

- Encode: English sentence.
  Decode1: Next sentence.
  Decode2: Previous sentence.



Query and nearest sentence
he ran his hand inside his coat , double-checking that the unopened letter was still there . he slipped his hand between his coat and his shirt , where the folded copies lay in a brown envelope .
im sure youll have a glamorous evening , she said , giving an exaggerated wink . im really glad you came to the party tonight , he said , turning to her .
although she could tell he had n't been too invested in any of their other chitchat , he seemed genuinely curious about this . although he had n't been following her career with a microscope , he 'd definitely taken notice of her appearances .
an annoying buzz started to ring in my ears , becoming louder and louder as my vision began to swim . a weighty pressure landed on my lungs and my vision blurred at the edges , threatening my consciousness altogether .
if he had a weapon , he could maybe take out their last imp , and then beat up errol and vanessa . if he could ram them from behind , send them sailing over the far side of the levee , he had a chance of stopping them .
then , with a stroke of luck , they saw the pair head together towards the portaloos . then , from out back of the house , they heard a horse scream probably in answer to a pair of sharp spurs digging deep into its flanks .
" i 'll take care of it , " goodman said , taking the phonebook . " i 'll do that , " julia said , coming in .
he finished rolling up scrolls and , placing them to one side , began the more urgent task of finding ale and tankards . he righted the table , set the candle on a piece of broken plate , and reached for his flint , steel , and tinder .

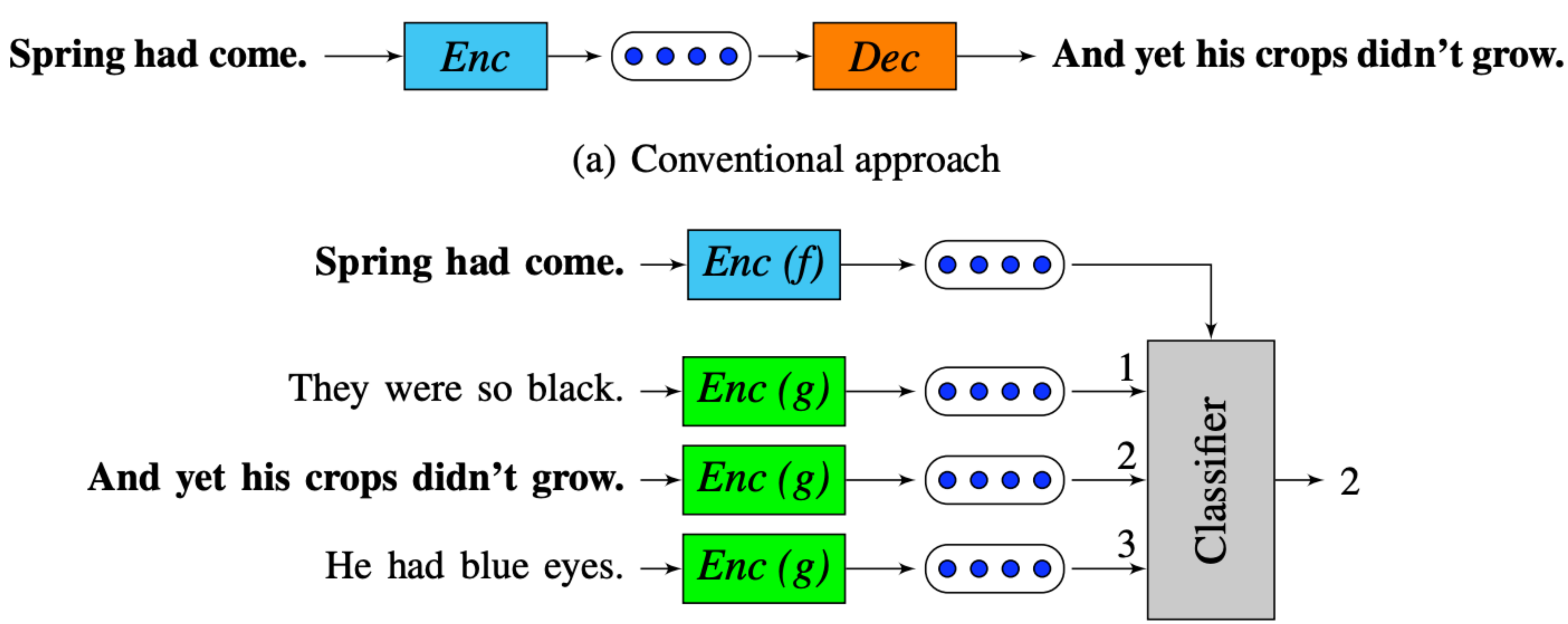
(what can we do with this similarity? can we tame it?)

Alternative Training

AN EFFICIENT FRAMEWORK FOR LEARNING SENTENCE REPRESENTATIONS

Lajanugen Logeswaran* **& Honglak Lee**†*
*University of Michigan, Ann Arbor, MI, USA
†Google Brain, Mountain View, CA, USA
{llajan,honglak}@umich.edu,honglak@google.com

discriminative classification



(b) Proposed approach

Alternative Training

another option: sentence order prediction

Encoder-Decoder with different modalities

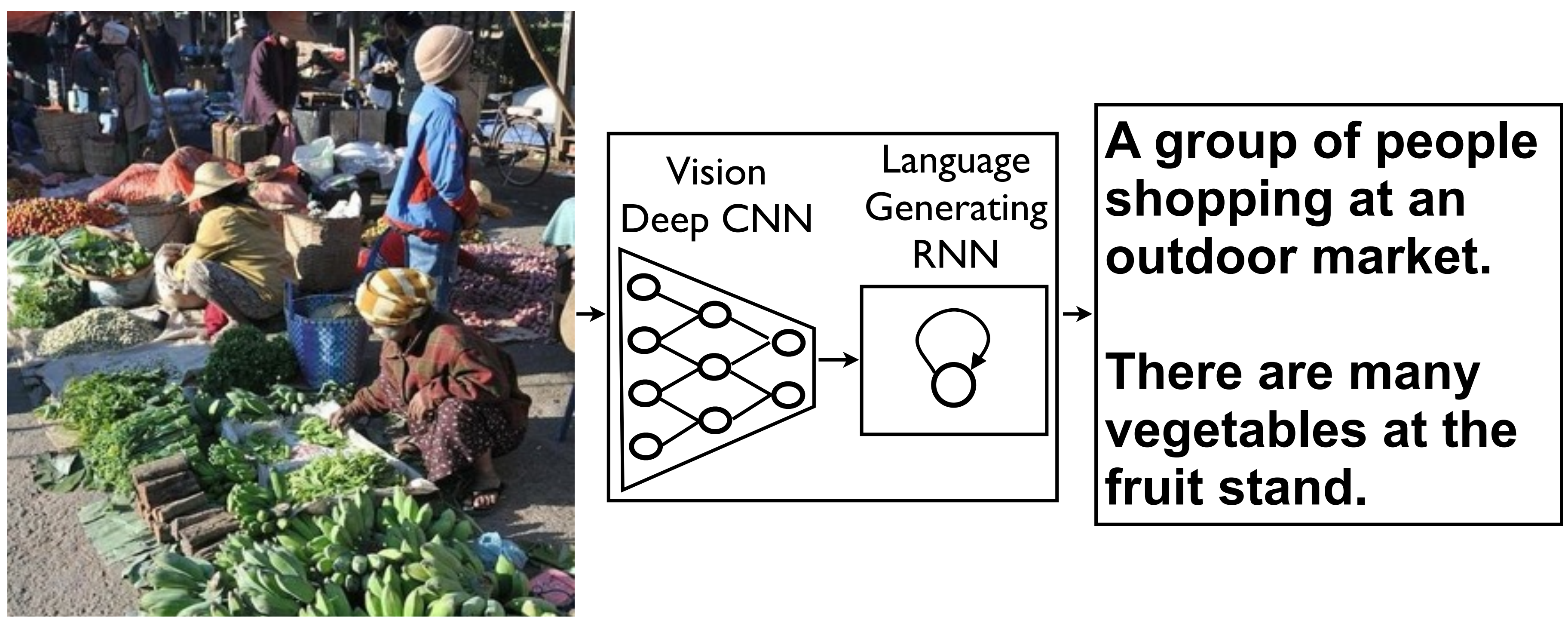
The encoded conditioning context need not be text, or even a sequence.

Encoder-Decoder with different modalities

Show and Tell: A Neural Image Caption Generator

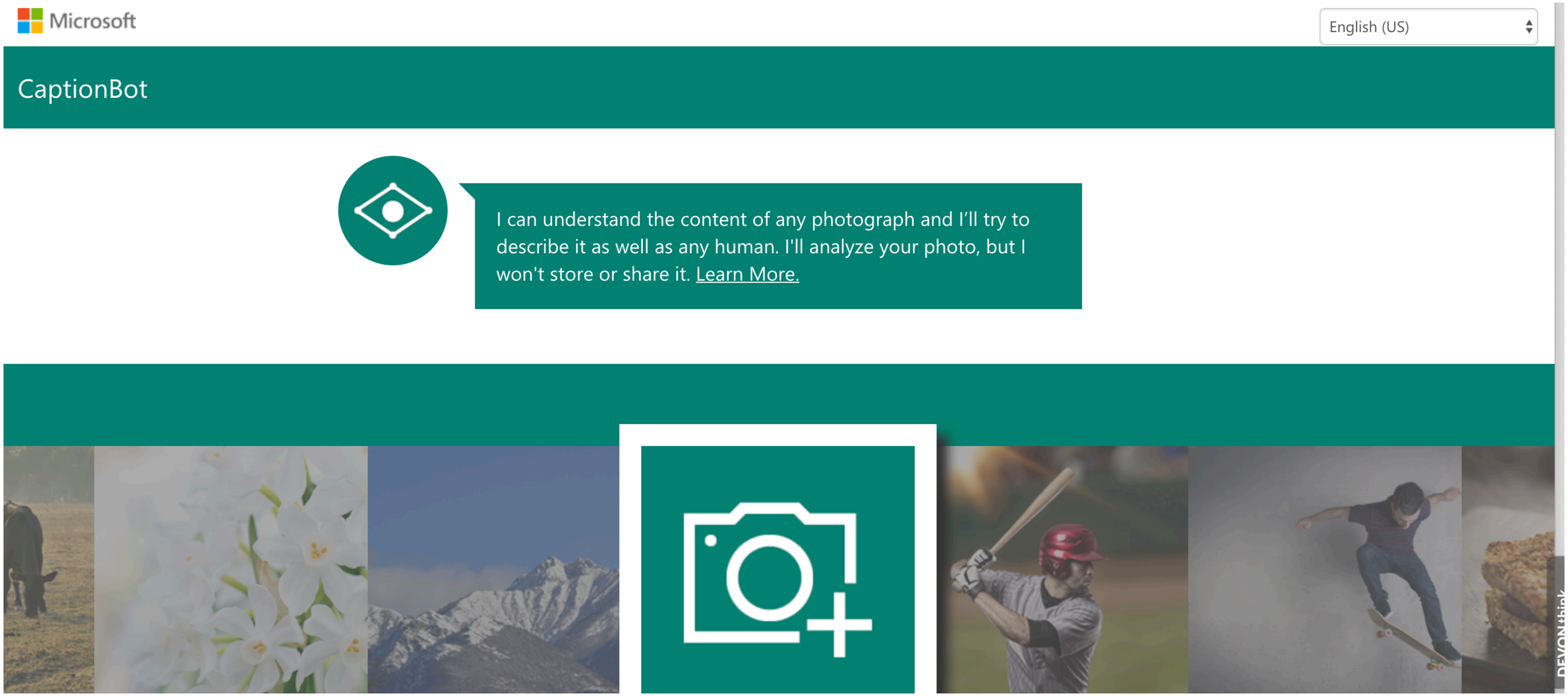
Oriol Vinyals, Google, vinyals@google.com
Alexander Toshev, Google, toshev@google.com
Samy Bengio, Google, bengio@google.com
Dumitru Erhan, Google, dumitru@google.com

- Encode: **image** to vector.
  Decode: a sentence describing the image.



- This sort-of works.
  In my opinion, looks more impressive than really is.

https://www.captionbot.ai/



I think it's a herd of cattle standing on top of a grass covered field.

I think it's a bench in front of a building.

I think it's a person in a grassy field.

I think it's a man in a business suit standing on a bench.

I am not really confident, but I think it's a man standing on a beach near the water.

I think it's a group of people sitting in front of a crowd.

I am not really confident, but I think it's a close up of a sheep.

Conditioned Generation: what's next?

- How can we encode interesting, structured conditions?
- ... and train them well?
- (this is more probably more of a data problem than an architecture problem. It is a very interesting problem though.)

Conditioned Generation Recap

- Read input into a vector.
- Learn to produce output based on encoded vector.
- Good when input/output have different lengths or different modalities.

Inverse problems

when one direction is easy but the other is very hard.

Generic simulation problem:

Given input x calculate outcome $y = F(x)$.

$$\begin{aligned} x \in X &: \quad \text{parameters / input} \\ y \in Y &: \quad \text{outcome / measurements} \\ F: X \to Y &: \quad \text{functional relation / model} \end{aligned}$$

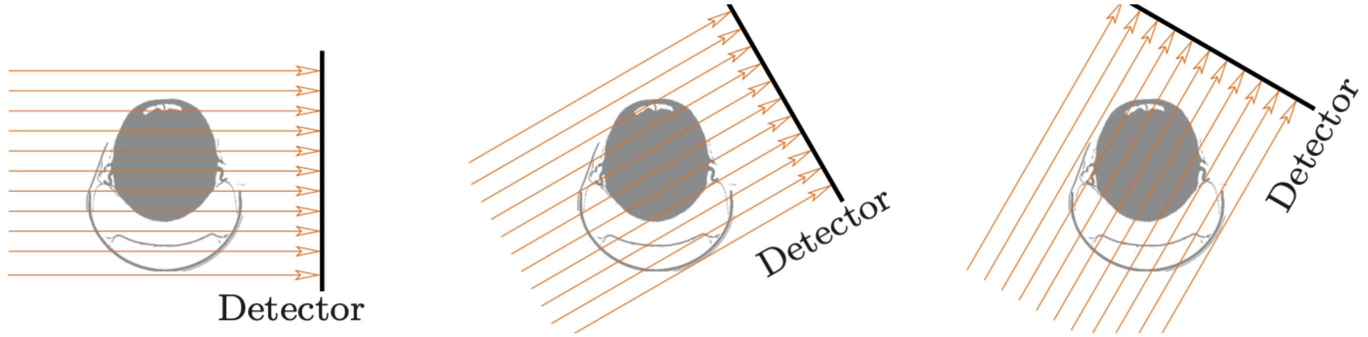
Goals:

- Prediction: Given x, calculate $y = F(x)$.
- Optimization: Find x, such that $F(x)$ is optimal.
- Inversion/Identification: Given $F(x)$, calculate x.

Computerized tomography

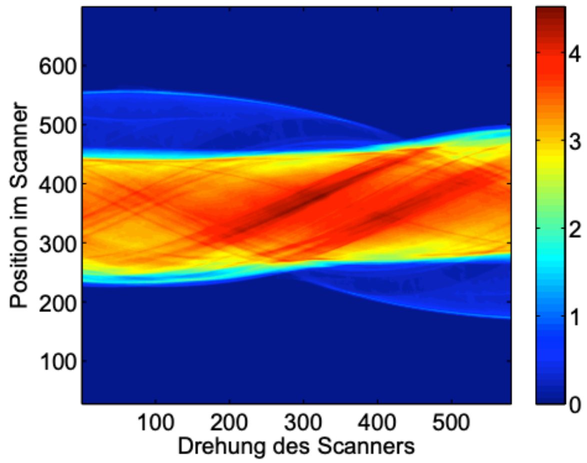
Nobel Prize in Physiology or Medicine 1979:
Allan M. Cormack and Godfrey N. Hounsfield

Idea: Take x-ray images from several directions



X-ray

$\overset{F}{\longmapsto}$



CT

Image

Measurements

https://www.math.uni-frankfurt.de/~harrach/talks/2015Seoul_Introduction_Inverse_Problems.pdf

Image deblurring

$\xrightarrow{F}$

x
True image

$y = F(x)$
Blurred image

Image coloring

https://github.com/jantic/DeOldify

https://www.math.uni-frankfurt.de/~harrach/talks/2015Seoul_Introduction_Inverse_Problems.pdf

Inverse Problems

- There are cases when we can perform a computation in one direction, but not the other.
- Opportunity for a deep learning approach!
    - Generate data in easy direction.
    - Train a model on the other direction.
    - For some symbolic tasks: can then **sample** from the model and **verify the correctness.**

Inverse Problems

- Examples of symbolic / verifiable cases:
  - **Easy:** C -> asm. **Hard:** asm -> C.
  - **Easy:** Derivatives. **Hard:** Integrals.
  - **Easy:** count things in text.
    Hard: generate fluent text that respects counts.

Inverse Problems

- Examples:
  - **Easy:** C -> asm. **Hard:** asm -> C.

Towards Neural Decompilation

Omer Katz
Technion
Israel
omerkatz@cs.technion.ac.il

Yuval Olshaker
Technion
Israel
olshaker@cs.technion.ac.il

Yoav Goldberg
Bar Ilan University
Israel
yoav.goldberg@gmail.com

Eran Yahav
Technion
Israel
yahave@cs.technion.ac.il

  - **Easy:** Derivatives. **Hard:** Integrals.

DEEP LEARNING FOR SYMBOLIC MATHEMATICS

Guillaume Lample*
Facebook AI Research
glample@fb.com

François Charton*
Facebook AI Research
fcharton@fb.com

  - **Easy:** count things in text.
    Hard: generate fluent text that respects counts.

Controlling Linguistic Style Aspects in Neural Language Generation

Jessica Ficler and **Yoav Goldberg**

Encoder abstractions.

The Encoder Abstractions

Take your input and transform / encode it.

Feed the encoded result to further processing.

The encoder is trained with the task.

The Encoder Abstractions

Three types of encoders:

- Symbol to vector. (lookup table, "embedding-layer")
- Sentence to vector (n to 1) [cbow, cnn+pooling, rnn]
- Sentence to vector-per-word (n to n) [cnn, bi-RNN]

The Encoder Abstractions

Three types of encoders:

- Symbol to vector. (lookup table, "embedding-layer")
- Sentence to vector (n to 1) [cbow, cnn+pooling, rnn]
  bi-RNN+pooling, bi-RNN+concat, ...
- Sentence to vector-per-word (n to n) [cnn, bi-RNN]

The Encoder Abstractions

Three types of encoders:

- Symbol to vector. (lookup table, "embedding-layer")
- Sentence to vector (n to 1) [cbow, cnn+pooling, rnn]
  bi-RNN+pooling, bi-RNN+concat, ...
- Sentence to vector-per-word (n to n) [cnn, bi-RNN]
  Transformer

The Encoder Abstractions

Three types of encoders:

- Symbol to vector. (lookup table, "embedding-layer")
- Sentence to vector (n to 1) [cbow, cnn+pooling, rnn]
  bi-RNN+pooling, bi-RNN+concat, ...
- Sentence to vector-per-word (n to n) [cnn, bi-RNN]
  Transformer

Training the encoders with the task is useful.
But what if we don't have enough data?
Can we train a "general" encoder?

The Encoder Abstractions

skip-thoughts

- Sentence to vector (n to 1) [cbow, cnn, rnn]
- Sentence to vector-per-word (n to n) [bi-RNN]

Transformer

elmo/bert

Language Model Pre-training

What if we have small training data?

- "self-supervised" training with language models.
- Main idea:

  train an encoder (RNN/Transformer/similar) on a language model (or similar) objective.

  use the resulting encoded representation (RNN states / Transformer vector(s)) as representations for the task you care about.

  (similar to pre-trained word-vectors, but here we are pre-training in-context encoders)

What if we have small training data?

- "self-supervised" training with language models.
- Main idea: **"transfer learning"**

  train an encoder (RNN/Transformer/similar) on a language model (or similar) objective.

  use the resulting encoded representation (RNN states / Transformer vector(s)) as representations for the task you care about.

  (similar to pre-trained word-vectors, but here we are pre-training in-context encoders)

Terminology

- "**static word embeddings**" --> produced by e.g. w2v
- "**contextualized word embeddings**" --> produced by, e.g., bi-RNN or Transformers
    - (note: training of bi-RNN, Transformers **also** produce "static" word embeddings (how?))

main papers (starting the trend)

Universal Language Model Fine-tuning for Text Classification

Jeremy Howard*
fast.ai
University of San Francisco
j@fast.ai

Sebastian Ruder*
Insight Centre, NUI Galway
Aylien Ltd., Dublin
sebastian@ruder.io

ULMfit

Deep contextualized word representations

Matthew E. Peters†, Mark Neumann†, Mohit Iyyer†, Matt Gardner†,
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark*, Kenton Lee*, Luke Zettlemoyer†*
{csquared,kentonl,lsz}@cs.washington.edu

†Allen Institute for Artificial Intelligence
*Paul G. Allen School of Computer Science & Engineering, University of Washington

ELMo

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

BERT

main papers (starting the trend)

Universal Language Model Fine-tuning for Text Classification

GPT, GPT2, GPT3

Jeremy Howard*
fast.ai
University of San Francisco
j@fast.ai

Sebastian Ruder*
Insight Centre, NUI Galway
Aylien Ltd., Dublin
sebastian@ruder.io

ULMfit

Deep contextualized word representations

Matthew E. Peters†, Mark Neumann†, Mohit Iyyer†, Matt Gardner†,
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark*, Kenton Lee*, Luke Zettlemoyer†*
{csquared,kentonl,lsz}@cs.washington.edu

†Allen Institute for Artificial Intelligence
*Paul G. Allen School of Computer Science & Engineering, University of Washington

ELMo

SpanBERT, XLNet, RoBERTA, ...

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Universal Language Model Fine-tuning for Text Classification

Jeremy Howard*
fast.ai
University of San Francisco
j@fast.ai

Sebastian Ruder*
Insight Centre, NUI Galway
Aylien Ltd., Dublin
sebastian@ruder.io

GPT, GPT2, GPT3

ULMfit

Deep contextualized word representations

Matthew E. Peters[†], Mark Neumann[†], Mohit Iyyer[†], Matt Gardner[†],
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark*, Kenton Lee*, Luke Zettlemoyer[†]*
{csquared,kentonl,lsz}@cs.washington.edu

[†]Allen Institute for Artificial Intelligence
*Paul G. Allen School of Computer Science & Engineering, University of Washington

ELMo

SpanBERT, XLNet, RoBERTA, ...

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

BERT

Next step:

BART

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

Mike Lewis*, Yinhan Liu*, Naman Goyal*, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer
Facebook AI
{mikelewis,yinhanliu,naman}@fb.com

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel* CRAFFEL@GMAIL.COM
Noam Shazeer* NOAM@GOOGLE.COM
Adam Roberts* ADAROB@GOOGLE.COM
Katherine Lee* KATHERINELEE@GOOGLE.COM
Sharan Narang SHARANNARANG@GOOGLE.COM
Michael Matena MMATENA@GOOGLE.COM
Yanqi Zhou YANQIZ@GOOGLE.COM
Wei Li MWEILI@GOOGLE.COM
Peter J. Liu PETERJLIU@GOOGLE.COM
Google, Mountain View, CA 94043, USA

T5

ULmfit, GPT, GPT2,3,...	Elmo/BERT	T5/BART
Language Model	Bidi-LM	Encode-Decode

ULmfit, GPT, GPT2,3,...	Elmo/BERT	T5/BART
Language Model	Bidi-LM	Encode-Decode
n to 1 **or** 0 to n	n to n	n to m
(depending on your p.o.v)		

ULMfit **(n to 1)**

- Train a strong general domain language model.
  - (3-layer LSTM, with good dropout, learning rate, optimizer, etc choices)

- Fine-tune the pre-trained LM on the in-domain data.
  - Two "tricks" to improve this part, see paper.

- Classify based on the LM states. Model:

$$LSTM(\mathbf{x_{1:n}}) = \mathbf{h_1}, ..., \mathbf{h_n} = \mathbf{H}$$
$$\tilde{\mathbf{h}} = [\mathbf{h_n}, \mathrm{maxpool}(\mathbf{H}), \mathrm{avgpool}(\mathbf{H})]$$
$$\hat{y} = softmax(MLP(\tilde{\mathbf{h}}))$$

Fine-tune the entire thing. (additional trick: gradual unfreezing)

ULMfit **(n to 1)**

- Works very well for classification.
- Can be easily adapted to bi-LSTM (how?)
- Why does it work?

GPT

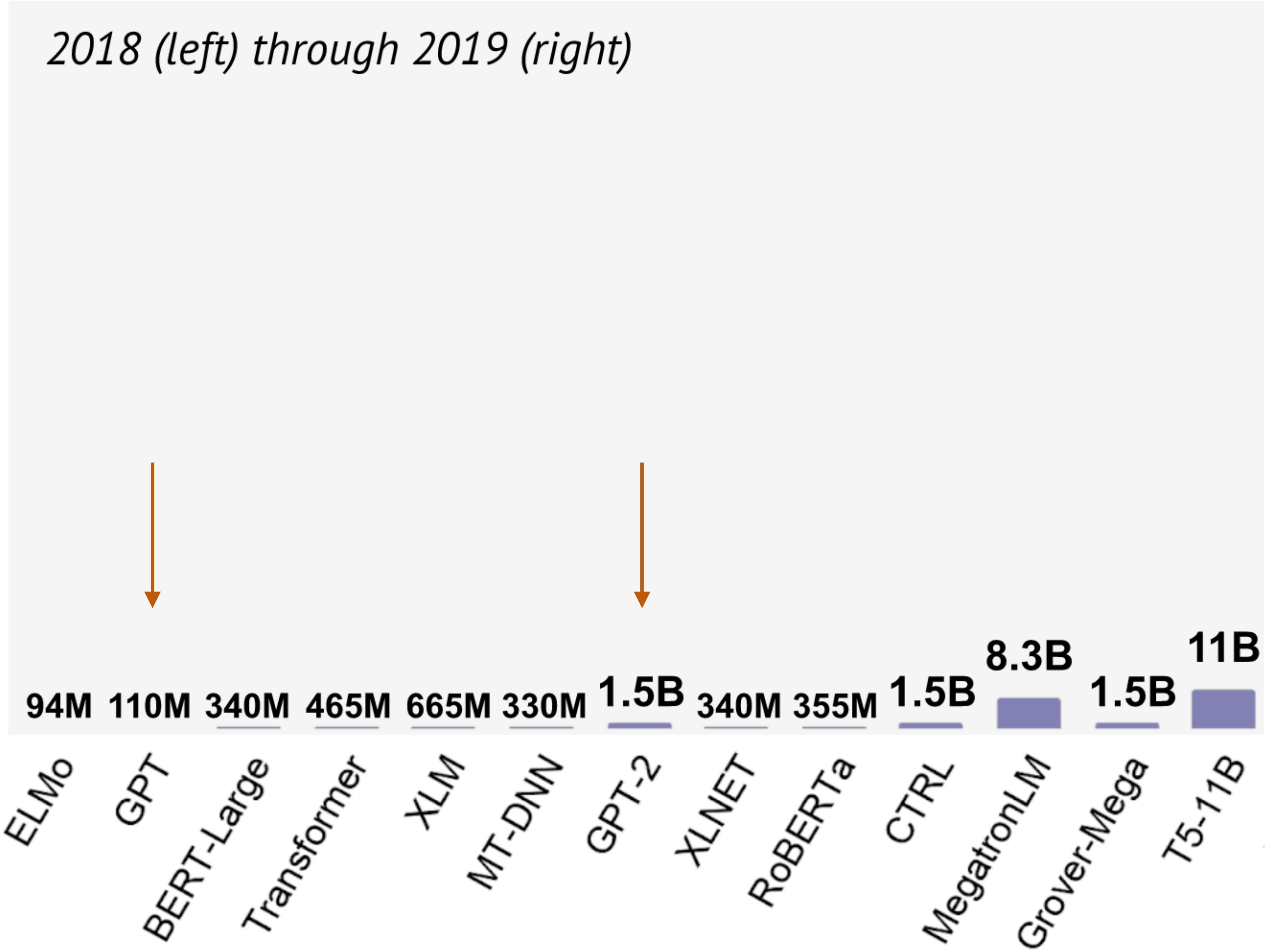
- Same idea as ULMfit, but with a transformer.
- (And, like any other LM, can also be used for generation)

GPT

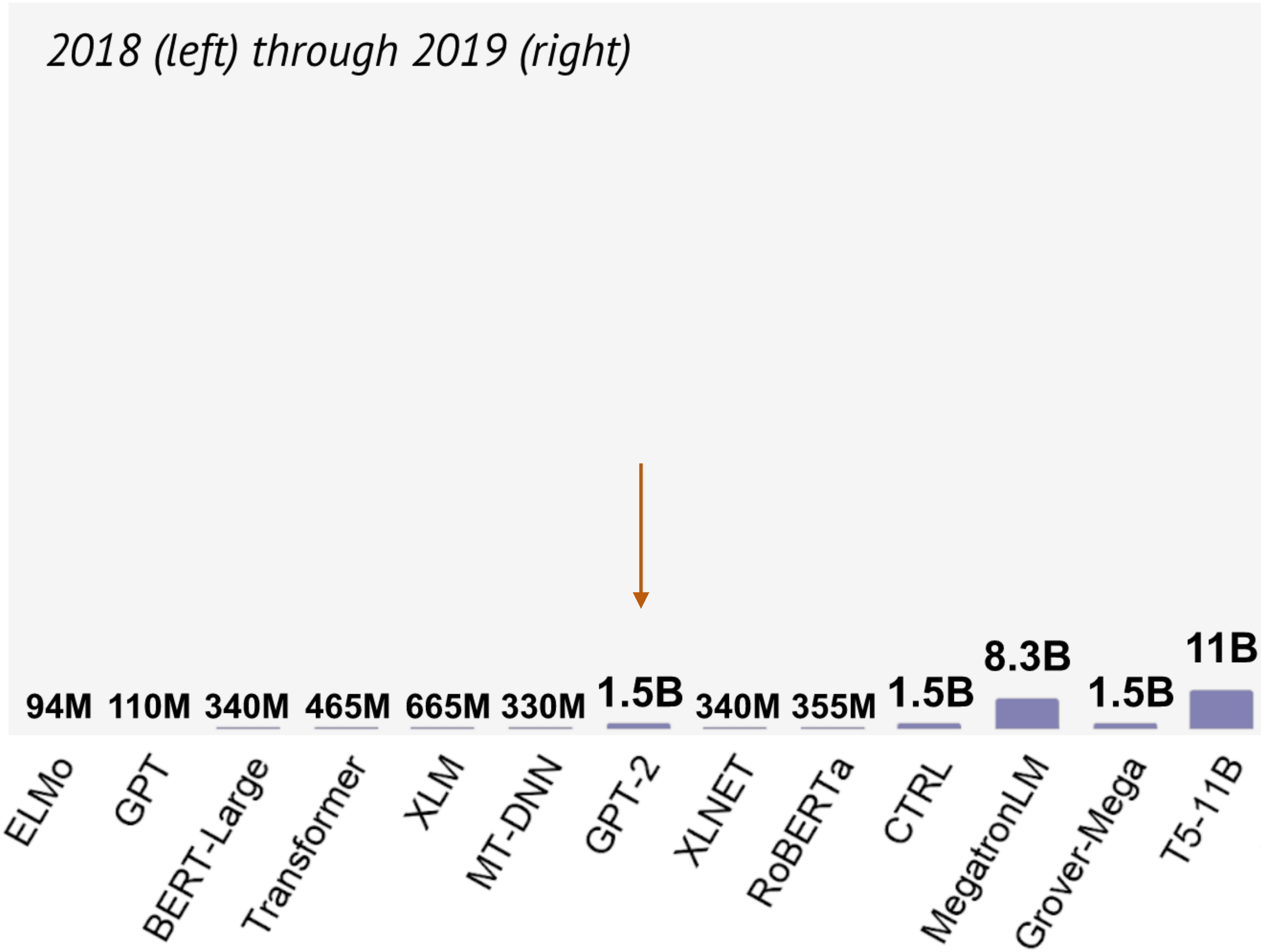
- Same idea as ULMfit, but with a transformer.
- (And, like any other LM, can also be used for generation)

GPT2, GPT3

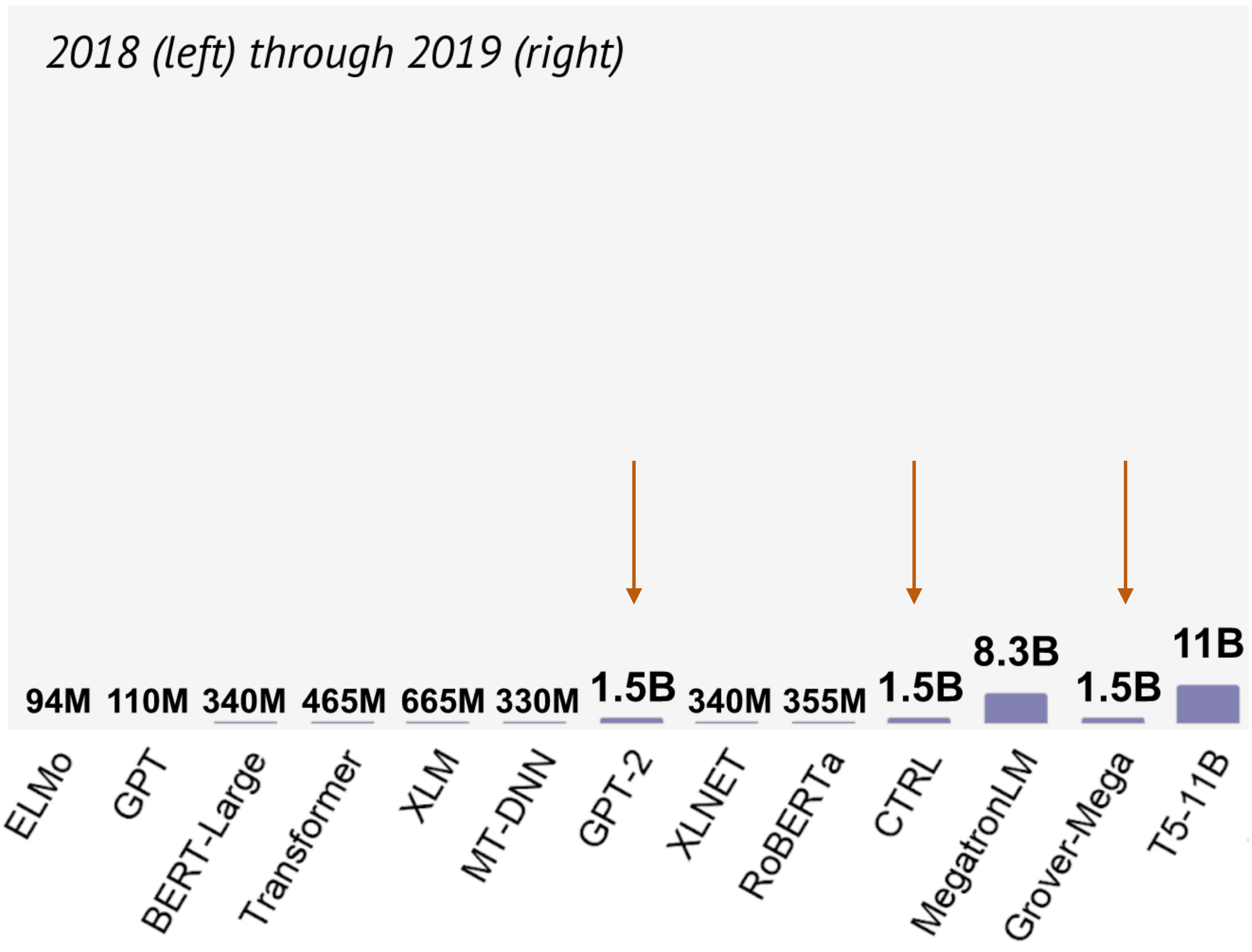
- Larger transformer (more layers, more heads, wider layers)
- More data



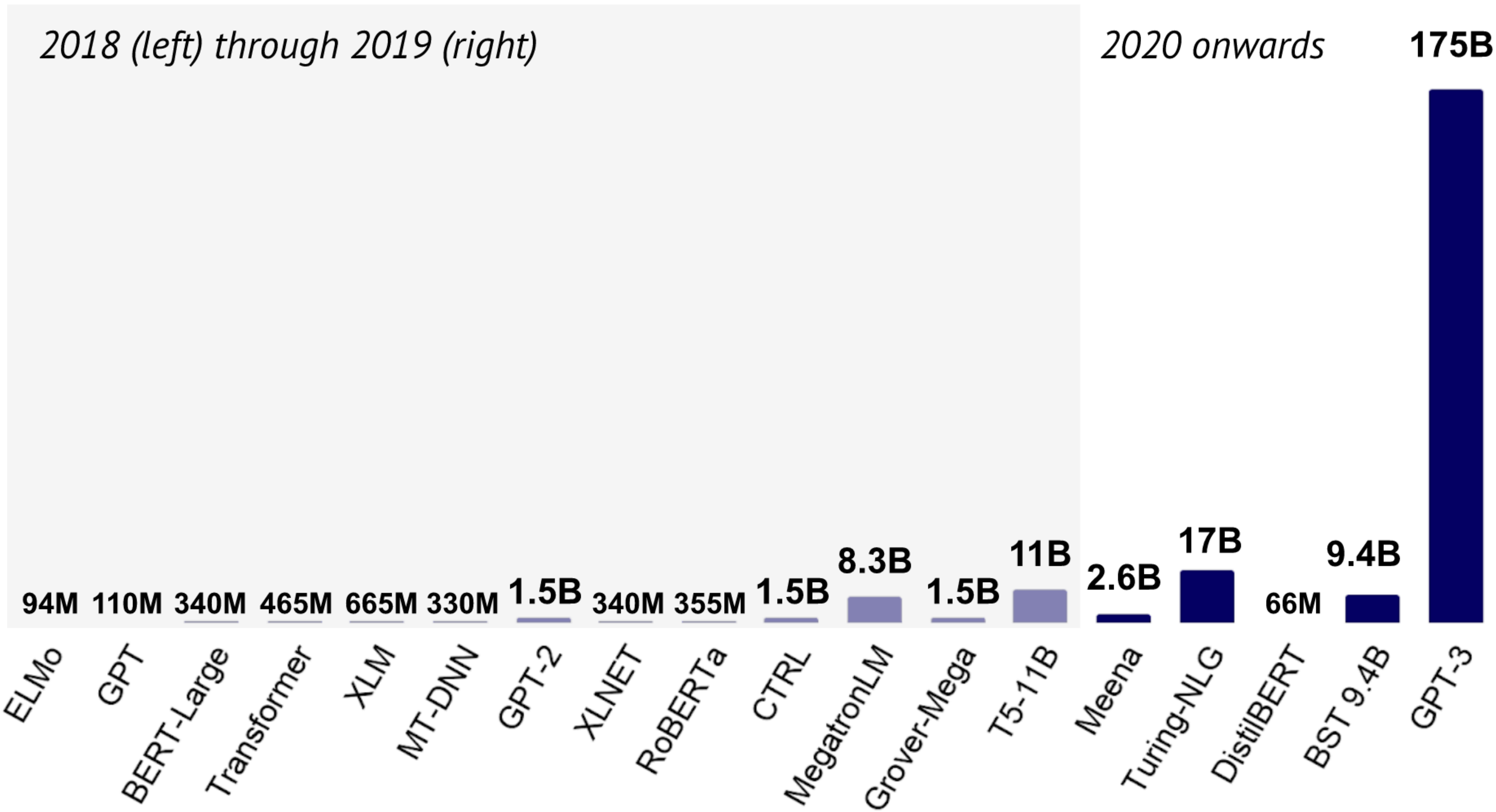
2018 (left) through 2019 (right)
94M
110M
340M
465M
665M
330M
1.5B
340M
355M
1.5B
8.3B
1.5B
11B
ELMo
GPT
BERT-Large
Transformer
XLM
MT-DNN
GPT-2
XLNET
RoBERTa
CTRL
MegatronLM
Grover-Mega
T5-11B



"too dangerous to release"



replications (sort-of) by others


2018 (left) through 2019 (right)
2020 onwards
94M
110M
340M
465M
665M
330M
1.5B
340M
355M
1.5B
8.3B
1.5B
11B
2.6B
17B
66M
9.4B
175B
ELMo
GPT
BERT-Large
Transformer
XLM
MT-DNN
GPT-2
XLNET
RoBERTa
CTRL
MegatronLM
Grover-Mega
T5-11B
Meena
Turing-NLG
DistilBERT
BST 9.4B
GPT-3

GPT2, GPT3

- Larger transformer (more layers, more heads, wider layers)

- More data

With the huge size of GPT3, we observe some "phase shift" in terms of abilities, in particular for learning from prompts.

GPT3 learns structures *very* well.
Does it also learn meaning?

GPT2, GPT3

- Larger transformer (more layers, more heads, wider layers)
- More data

With the huge size of GPT3, we observe some "phase shift" in terms of abilities, in particular for learning from prompts.

GPT3 learns structures *very* well.
Does it also learn meaning?

Climbing towards NLU:
On Meaning, Form, and Understanding in the Age of Data

Emily M. Bender
University of Washington
Department of Linguistics
ebender@uw.edu

Alexander Koller
Saarland University
Dept. of Language Science and Technology
koller@coli.uni-saarland.de

Provable Limitations of Acquiring Meaning from Ungrounded Form:
What will Future Language Models Understand?

William Merrill[*] **Yoav Goldberg**[*,†] **Roy Schwartz**[‡] **Noah A. Smith**[*,§]
[*] Allen Institute for AI [†] Bar Ilan University
[‡] Hebrew University of Jerusalem [§] University of Washington
{willm,roys,yoavg,noah}@allenai.org

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

Emily M. Bender*
ebender@uw.edu
University of Washington
Seattle, WA, USA

Timnit Gebru*
timnit@blackinai.org
Black in AI
Palo Alto, CA, USA

Angelina McMillan-Major
aymm@uw.edu
University of Washington
Seattle, WA, USA

Shmargaret Shmitchell
shmargaret.shmitchell@gmail.com
The Aether

(not a very interesting question)

ELMo / BERT **(n to n)**

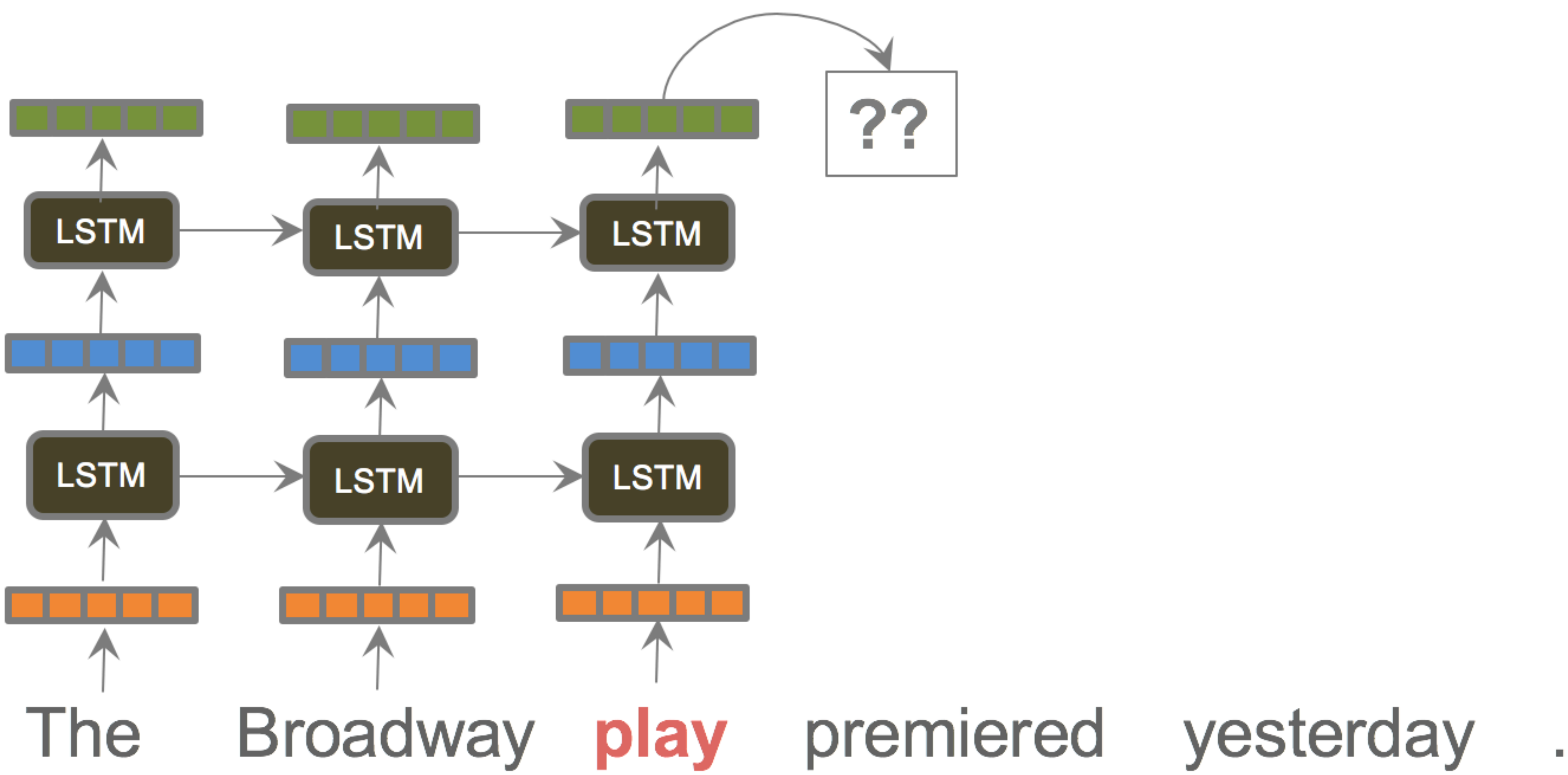
- Not just classification.

  We **replace word embeddings** with **contextualized word vectors**.

- Each **word** is represented as its encoded state. The in-context word vectors are then fed to further tasks.

ELMo

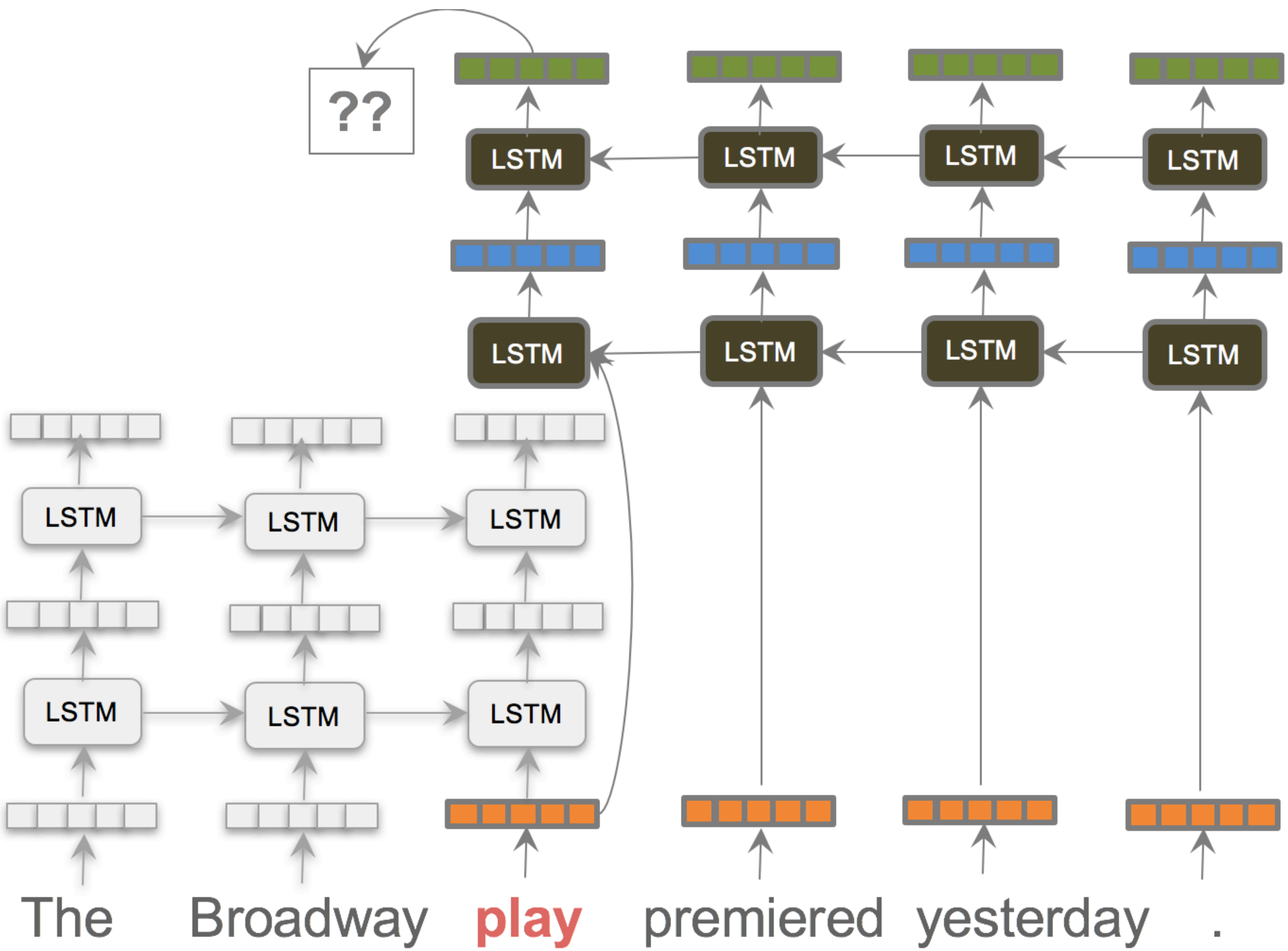
bi-LM



ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

ELMo

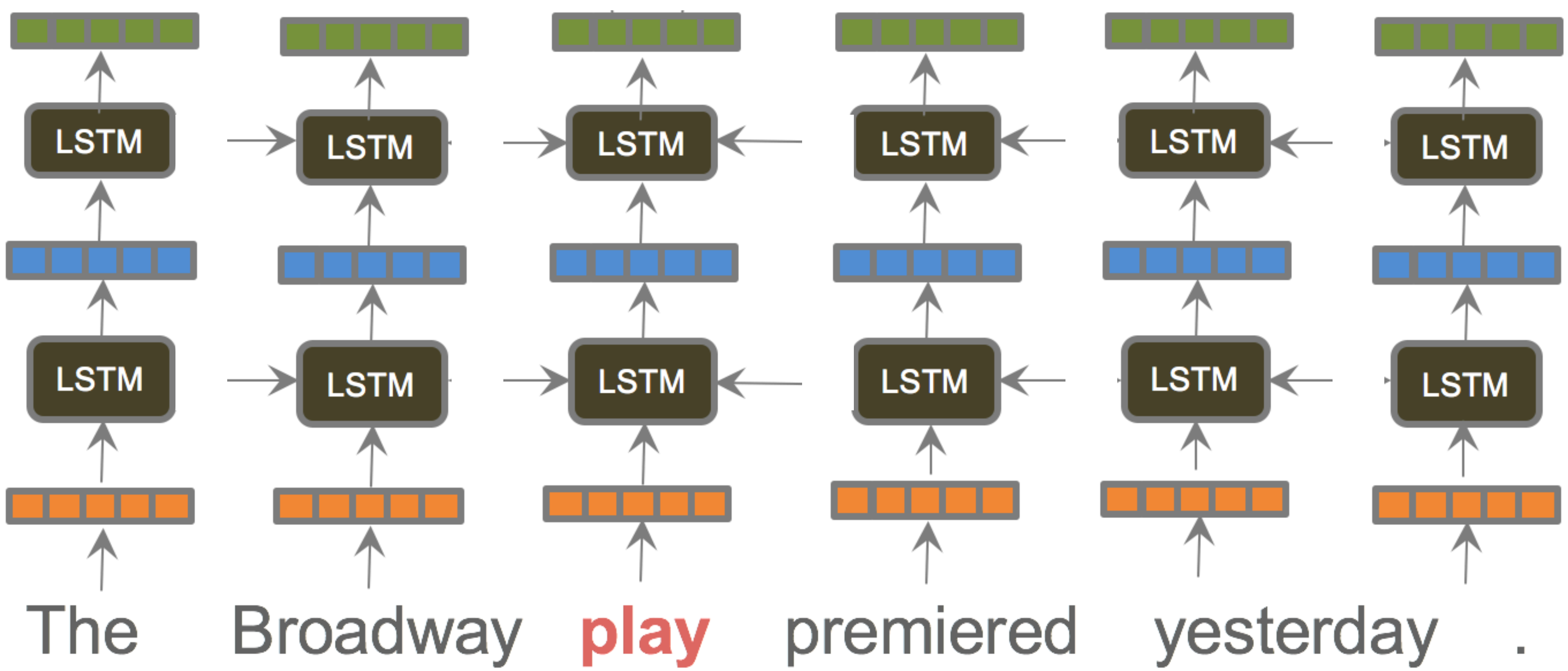
bi-LM



ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

ELMo

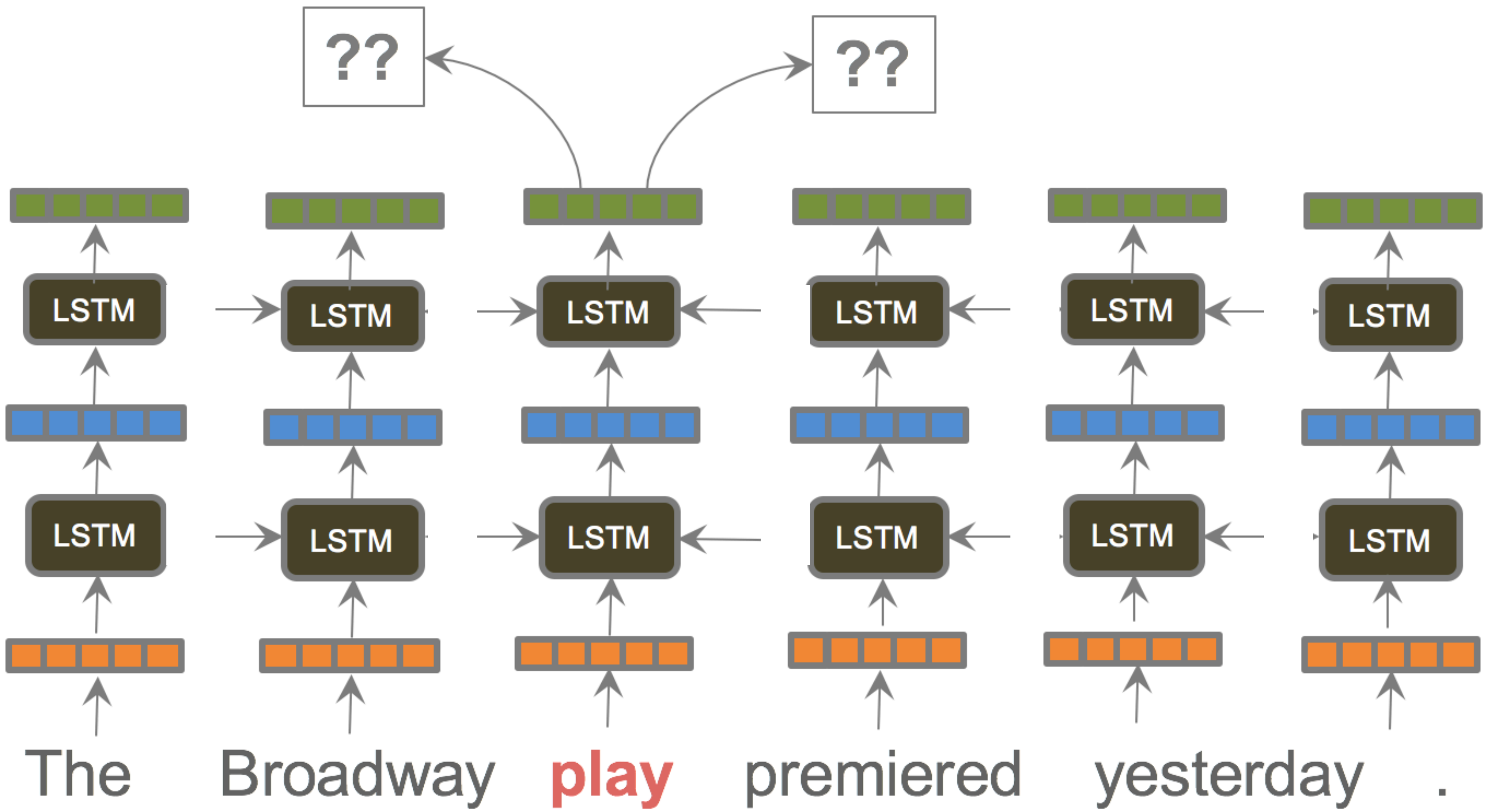
bi-LM



LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
The
Broadway
play
premiered
yesterday
.

ELMo

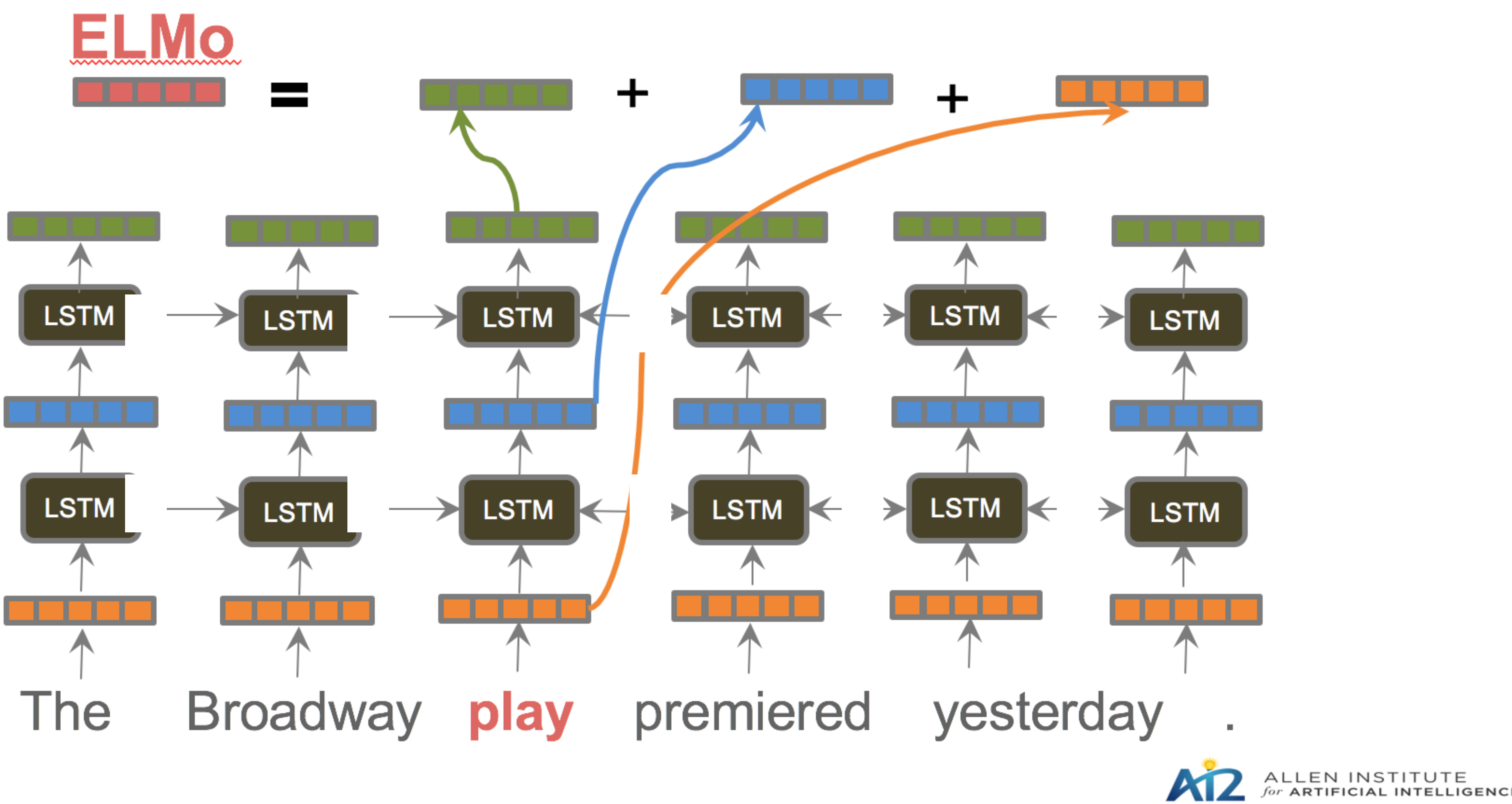
bi-LM



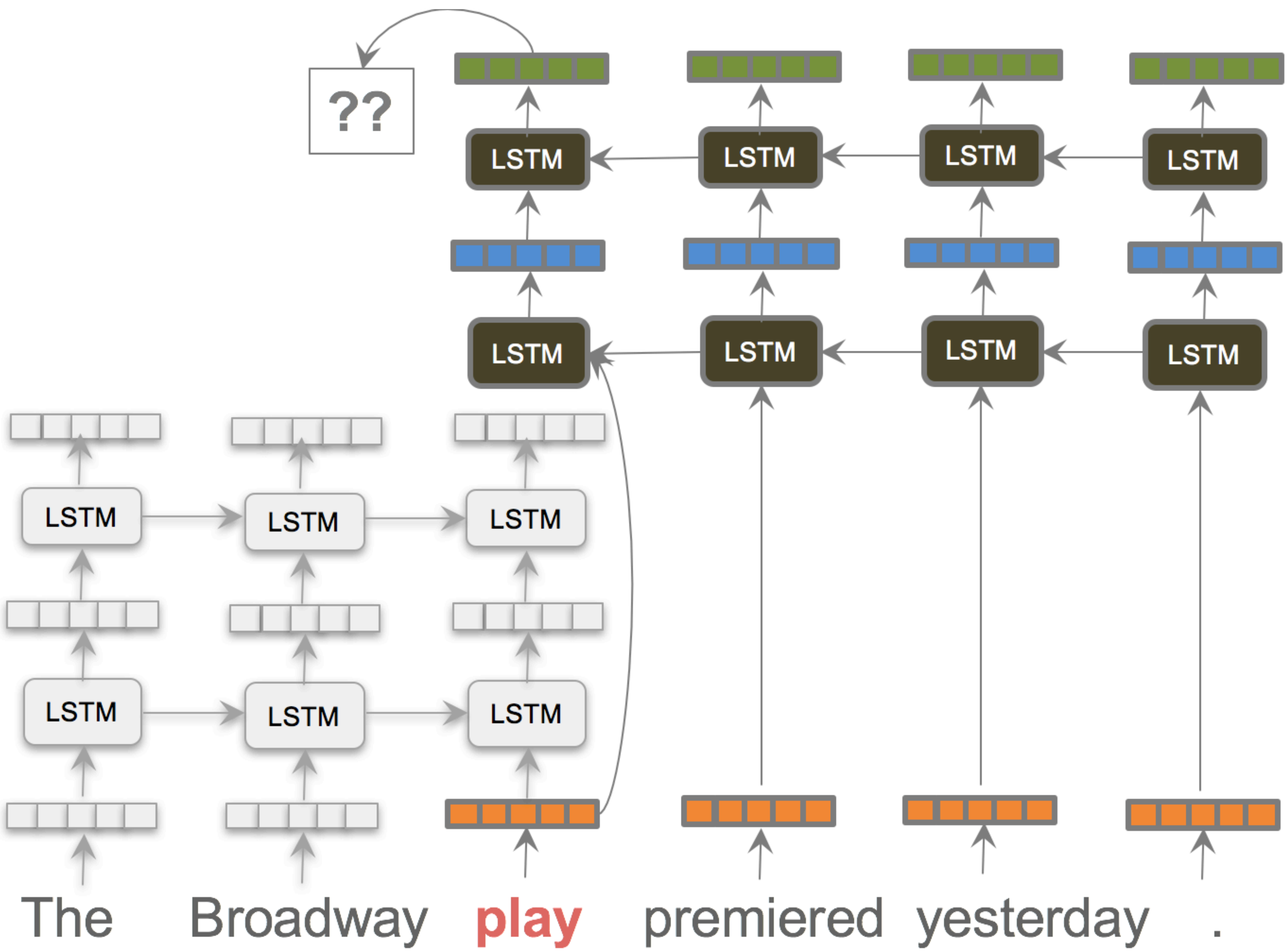
??
??
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
The
Broadway
play
premiered
yesterday
.

ELMo

bi-LM

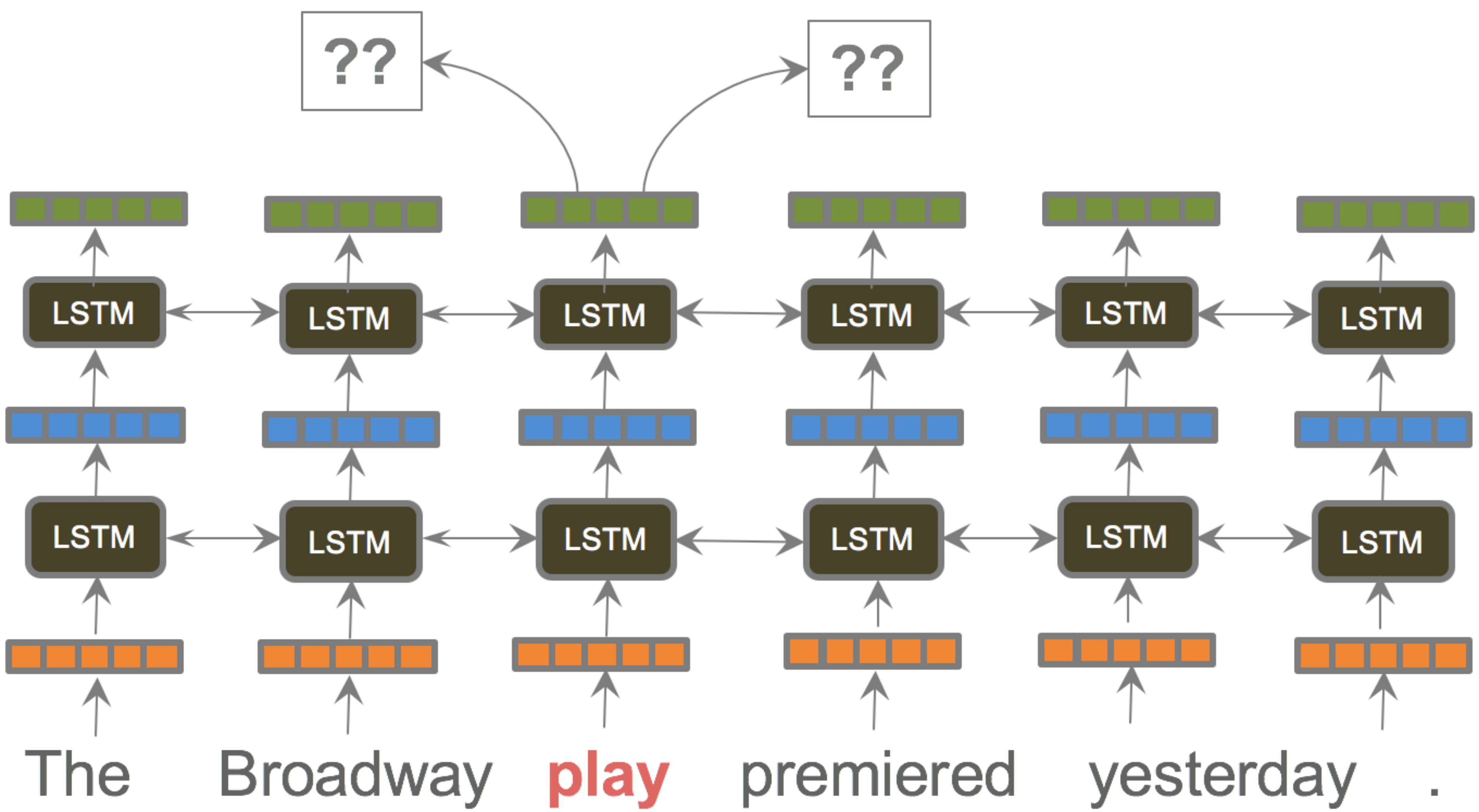

ELMo
=
+
+
LSTM
The
Broadway
play
premiered
yesterday
.
ALLEN INSTITUTE
for ARTIFICIAL INTELLIGENCE

Note: bi-LM (vs bi-RNN)
Two **separate** LMs



Note: bi-LM.
Two separate LMs

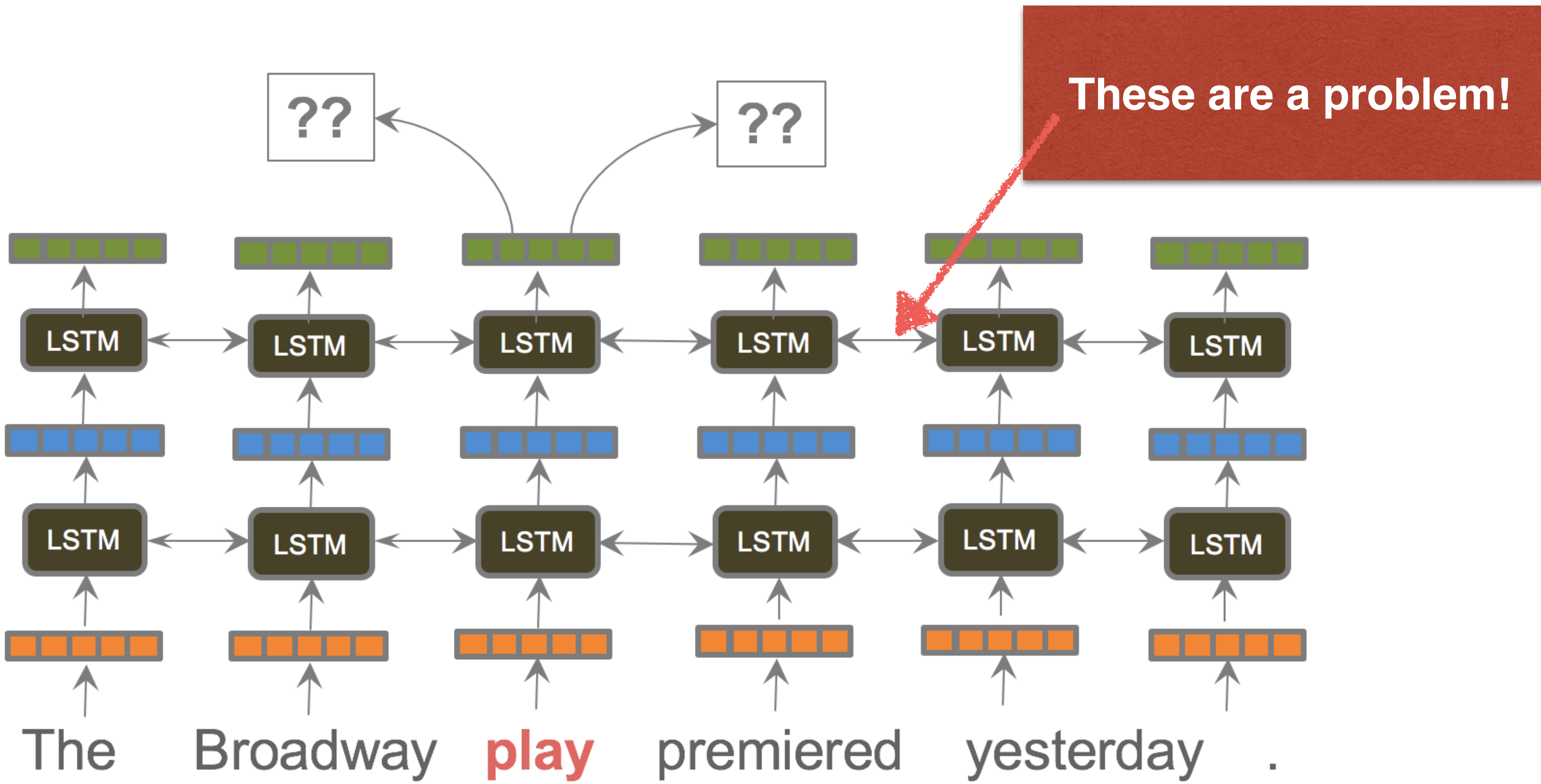
full deep bi-LSTM LM will not work. (why?)



ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

Note: bi-LM. Two separate LMs

full deep bi-LSTM LM will not work. (why?)



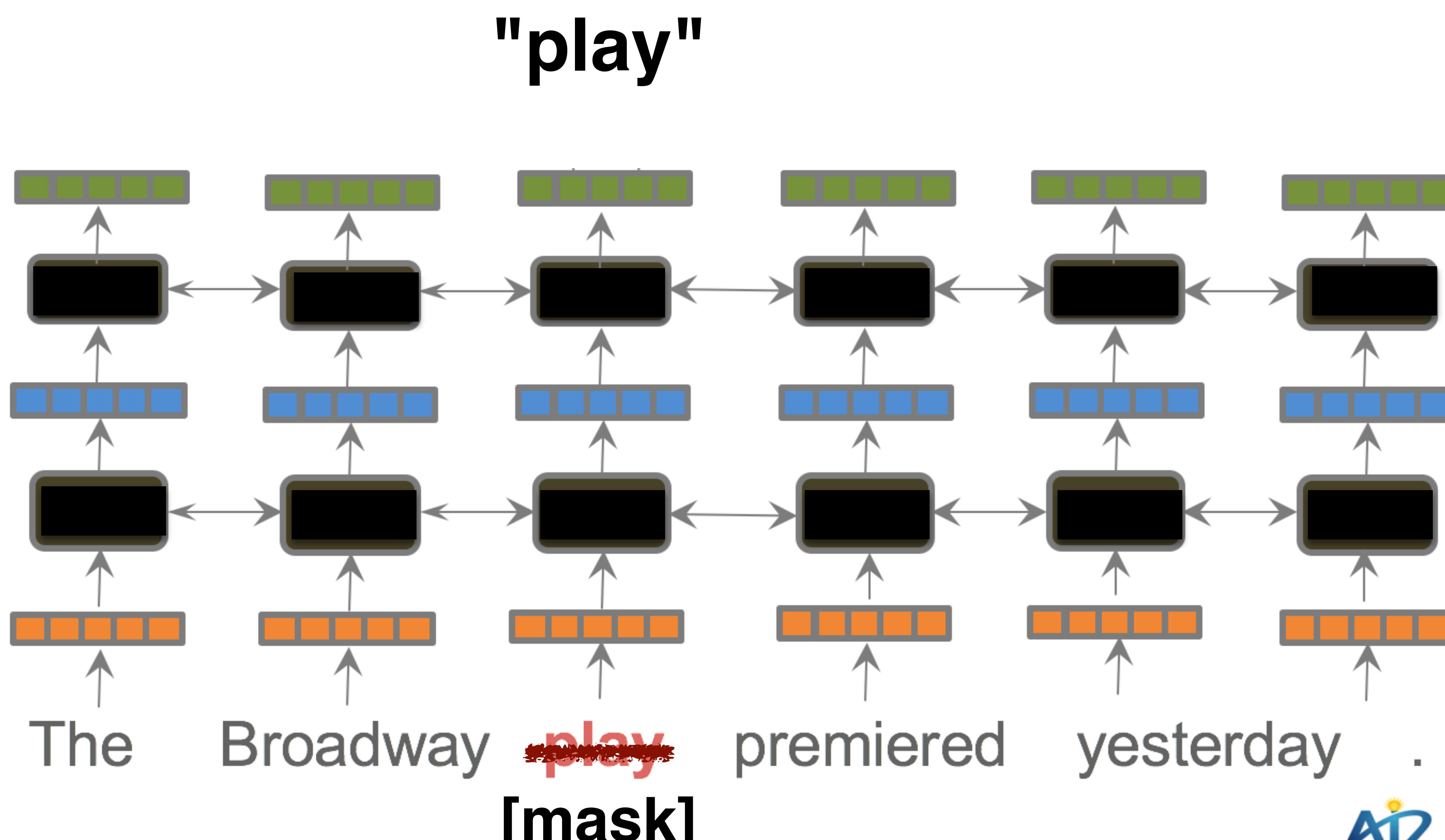
ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

BERT

- Several modifications:
- (1) LSTM --> Transformer
- (2) additional "skip-thought"-like objective
  (next sentence prediction)
- **(3) Real bidirectional+deep model.**
  - **with a masked-LM**

BERT

real deep bidirectional with masked-LM

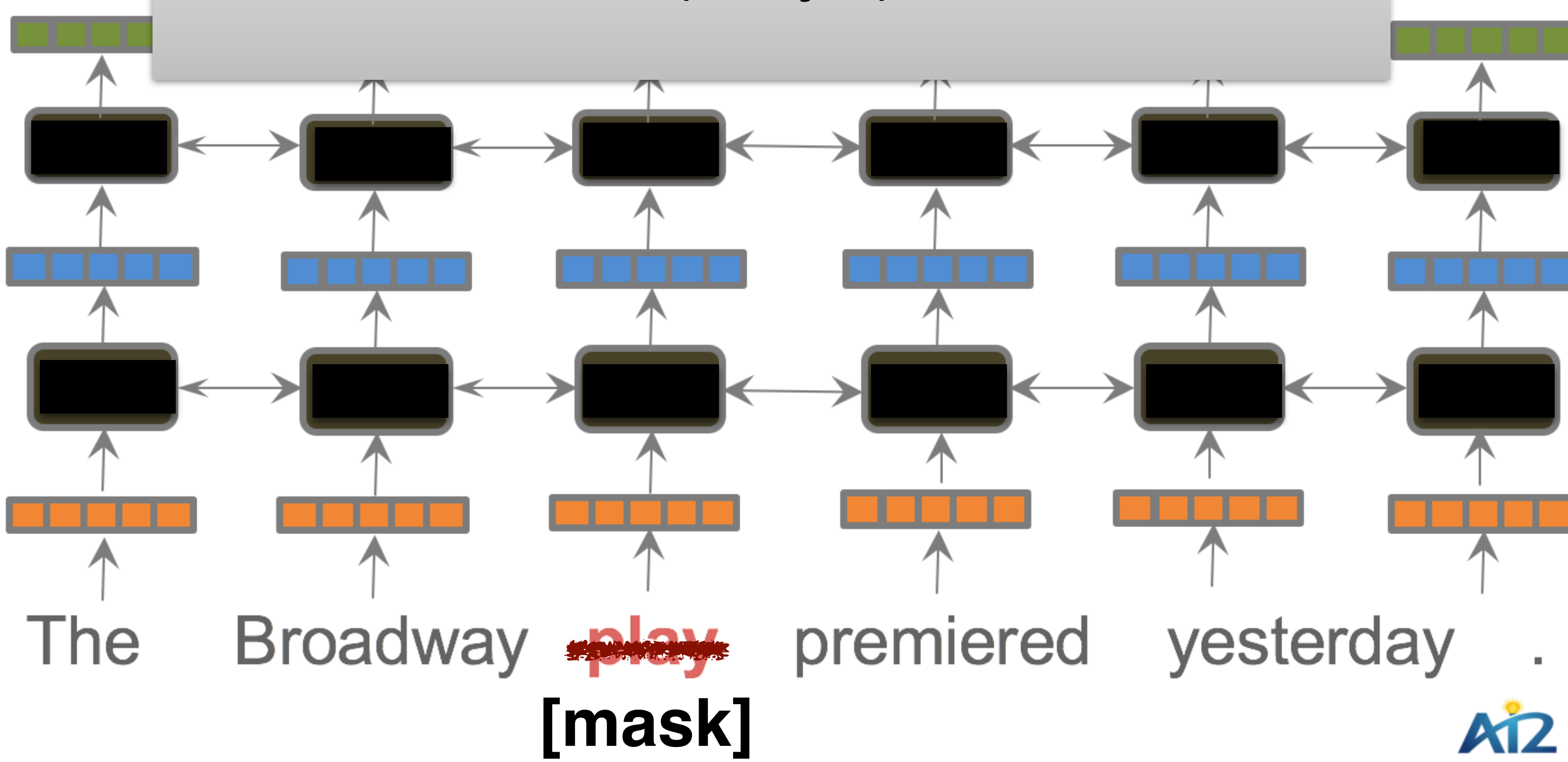


BERT

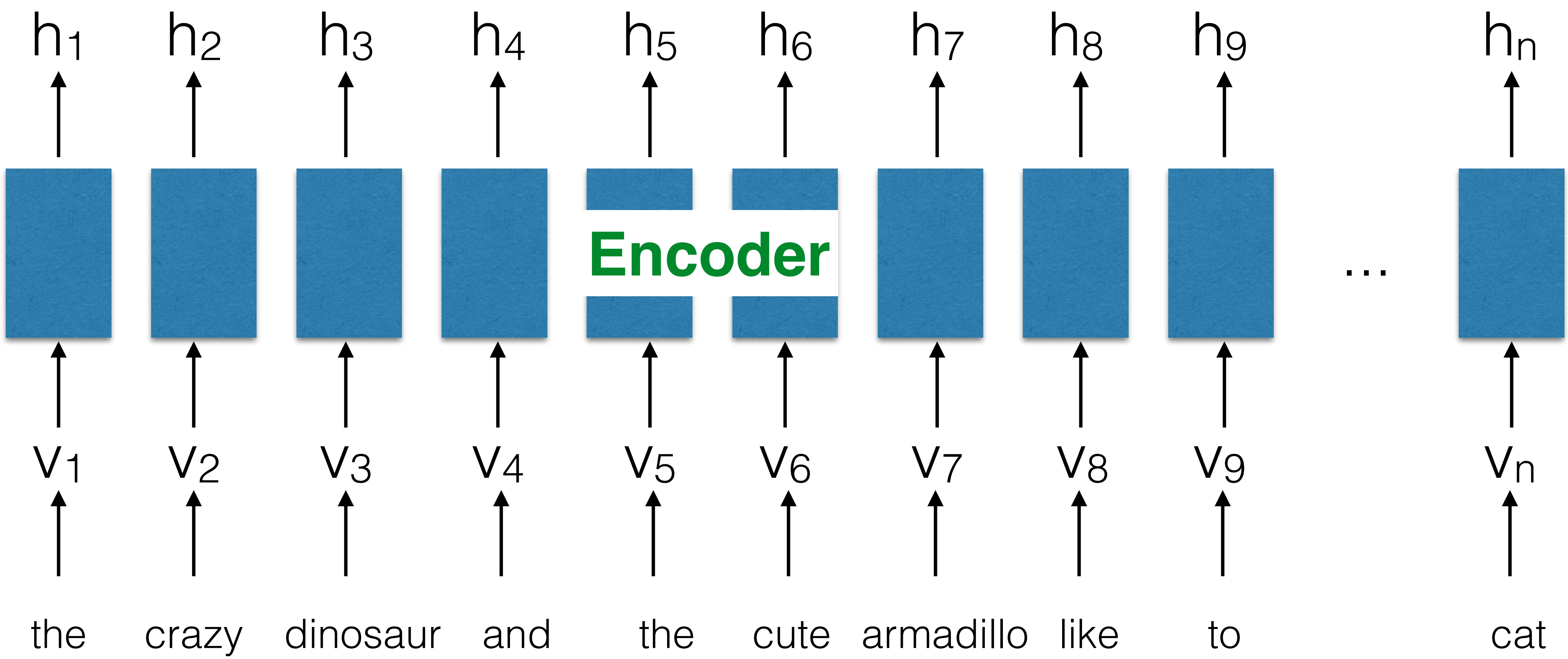
real deep bidirectional with masked-LM

Much more expensive to train!

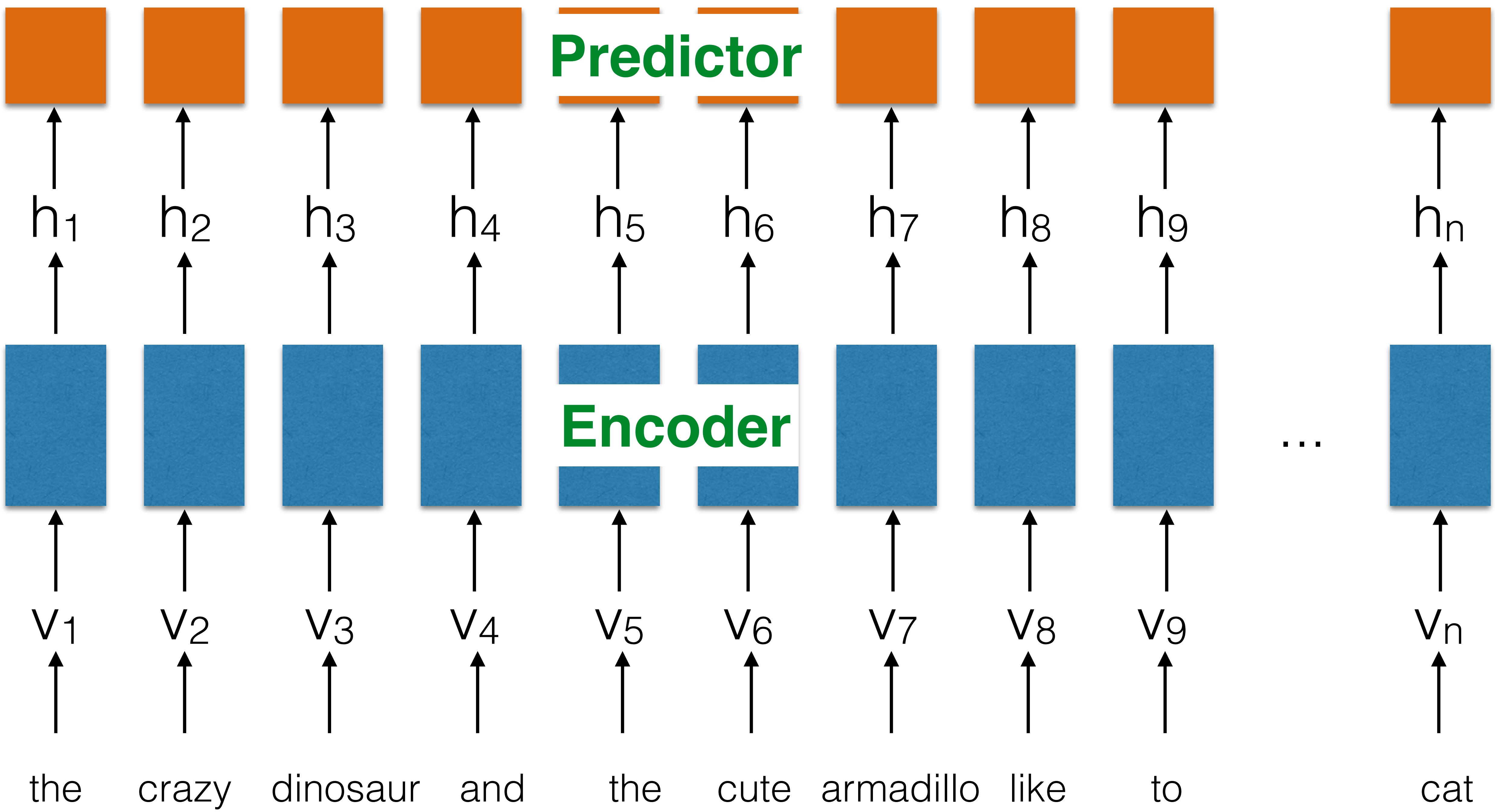
(why?)



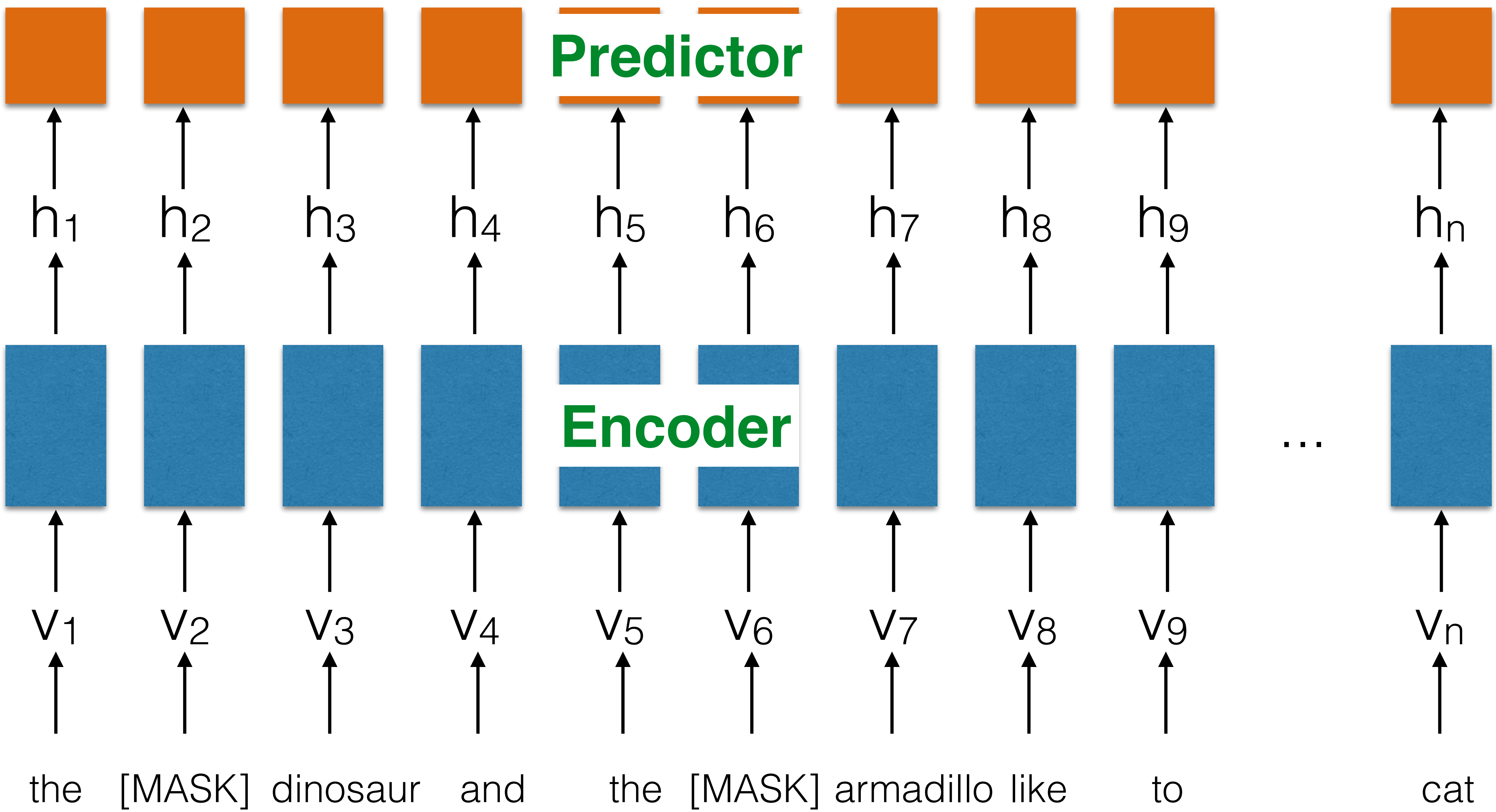
ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE



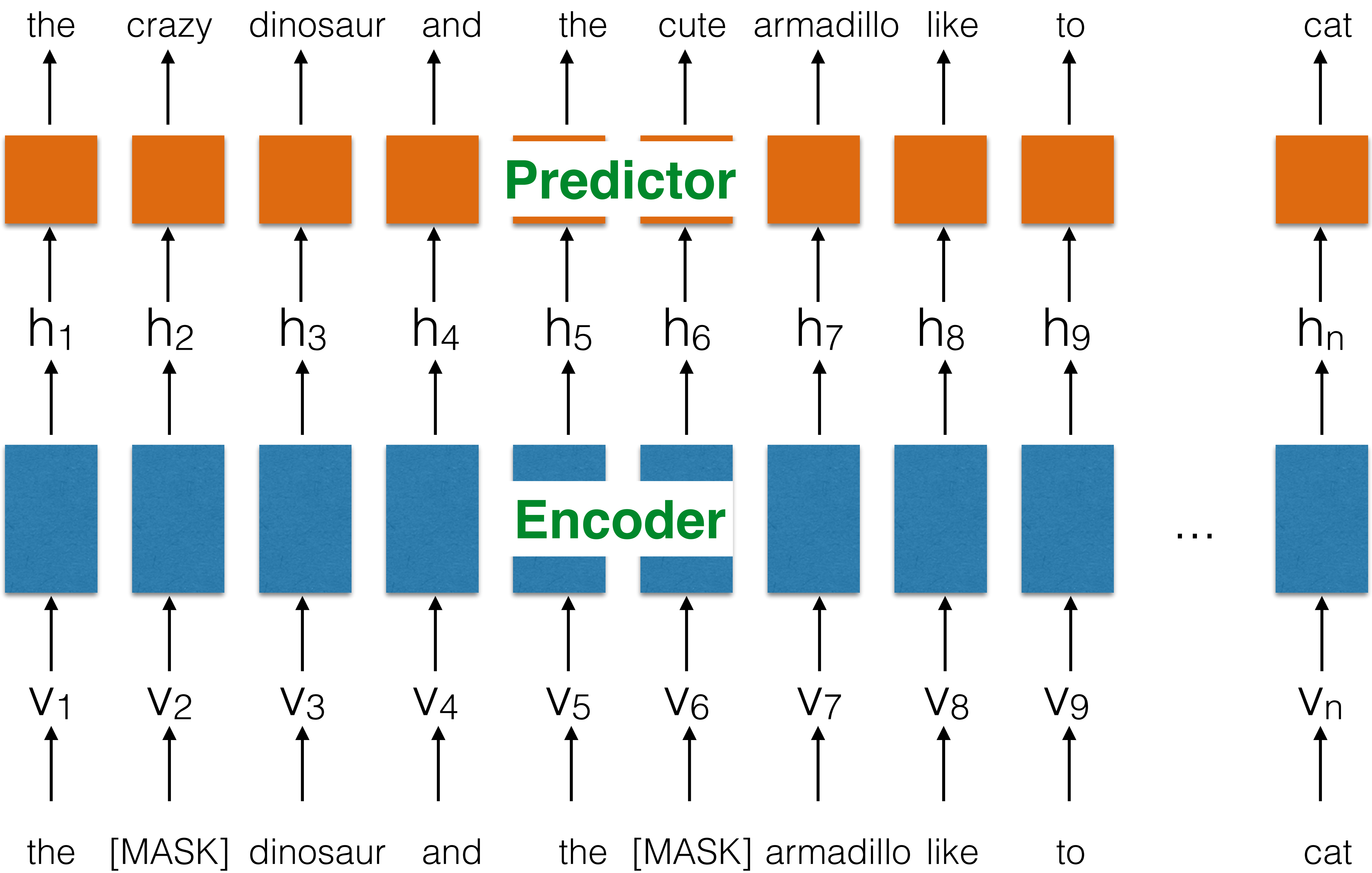
h1 h2 h3 h4 h5 h6 h7 h8 h9 ... hn
Encoder
v1 v2 v3 v4 v5 v6 v7 v8 v9 vn
the crazy dinosaur and the cute armadillo like to cat


Predictor
h1 h2 h3 h4 h5 h6 h7 h8 h9 hn
Encoder
...
v1 v2 v3 v4 v5 v6 v7 v8 v9 vn
the crazy dinosaur and the cute armadillo like to cat

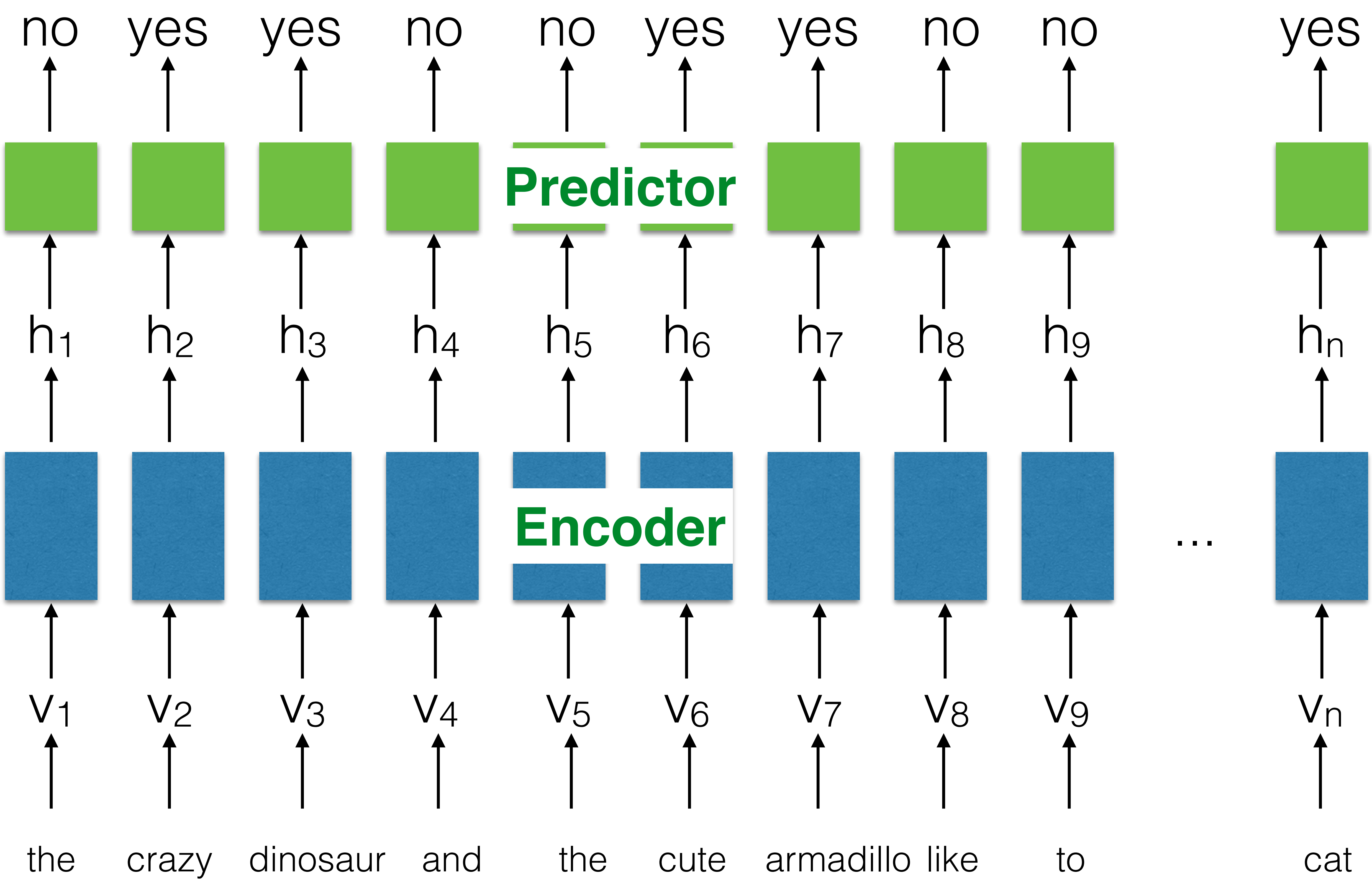
at (pre)train time



at (pre)train time

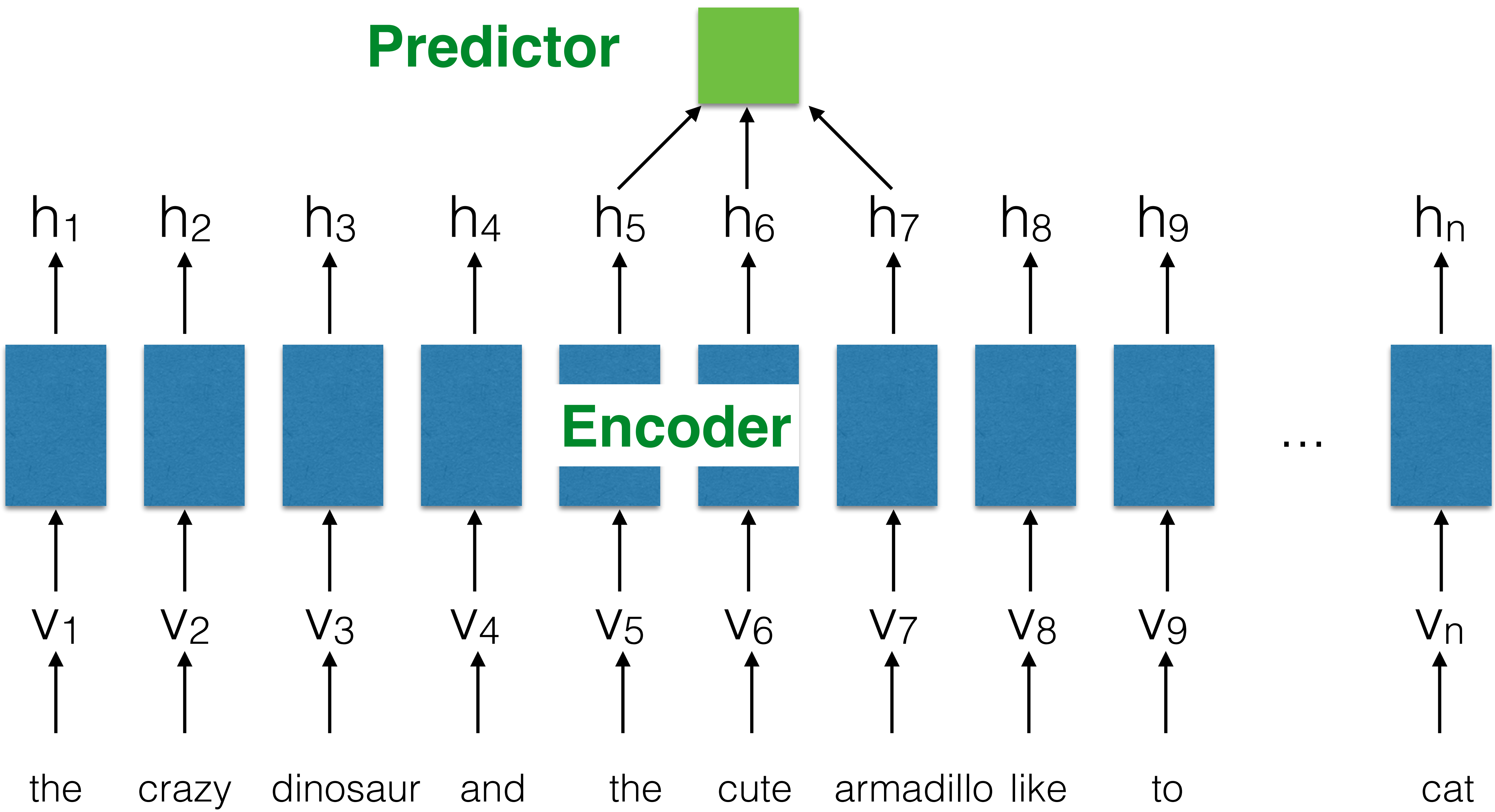


at fine-tuning / test time (different predictor, new task)

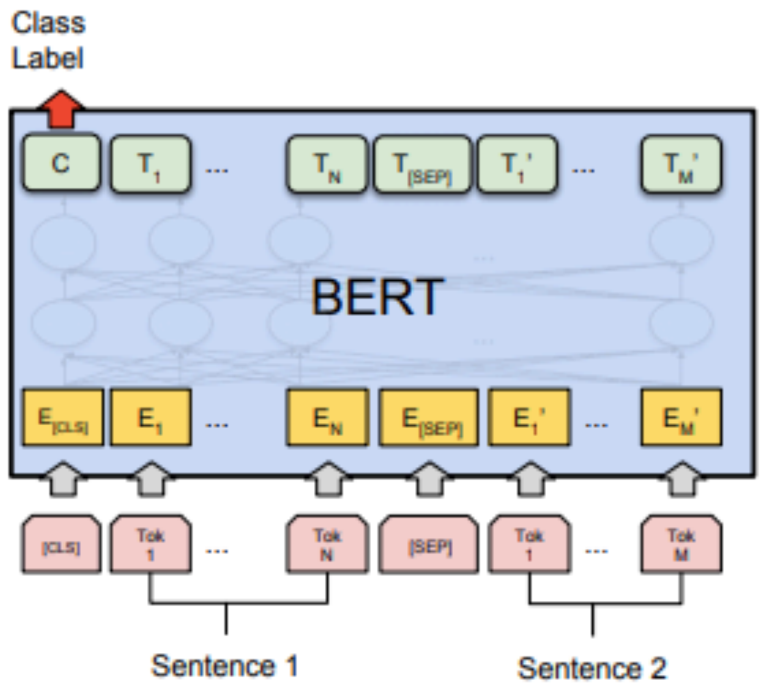


at fine-tuning / test time (different predictor, new task)

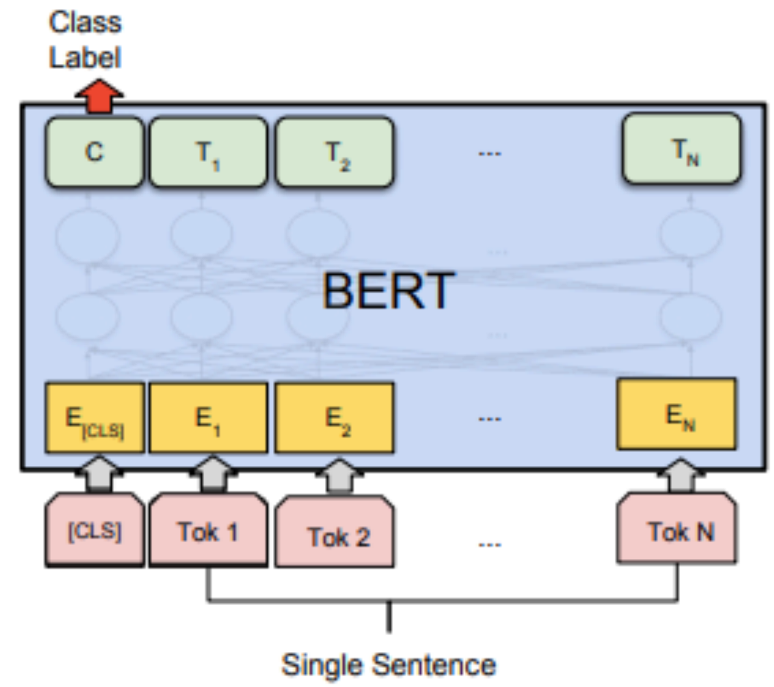
may predict based on sub-sequences



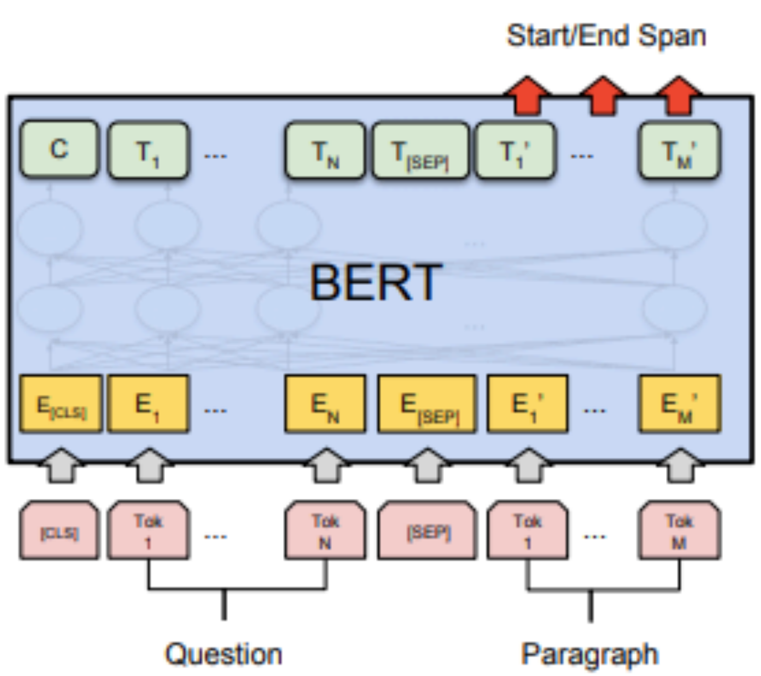
Using pre-trained encoders



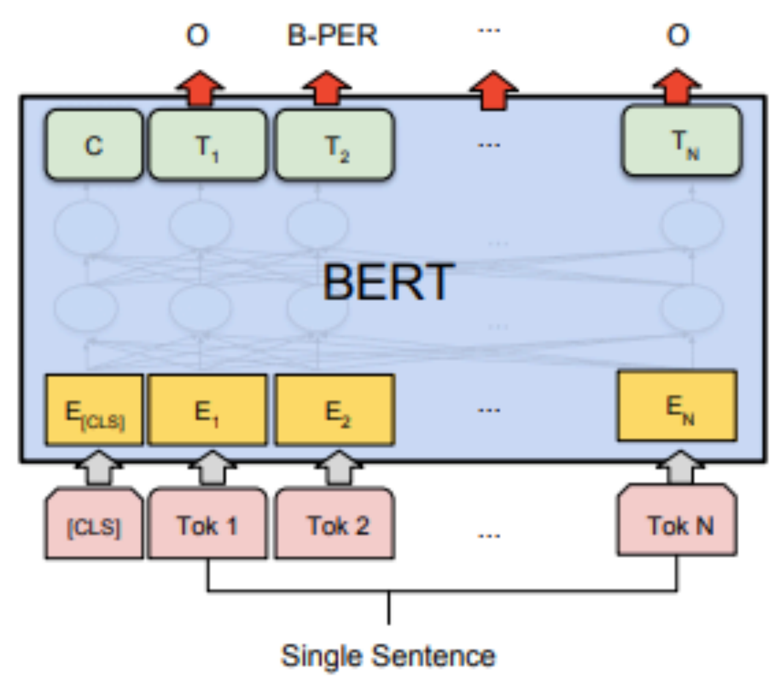
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA

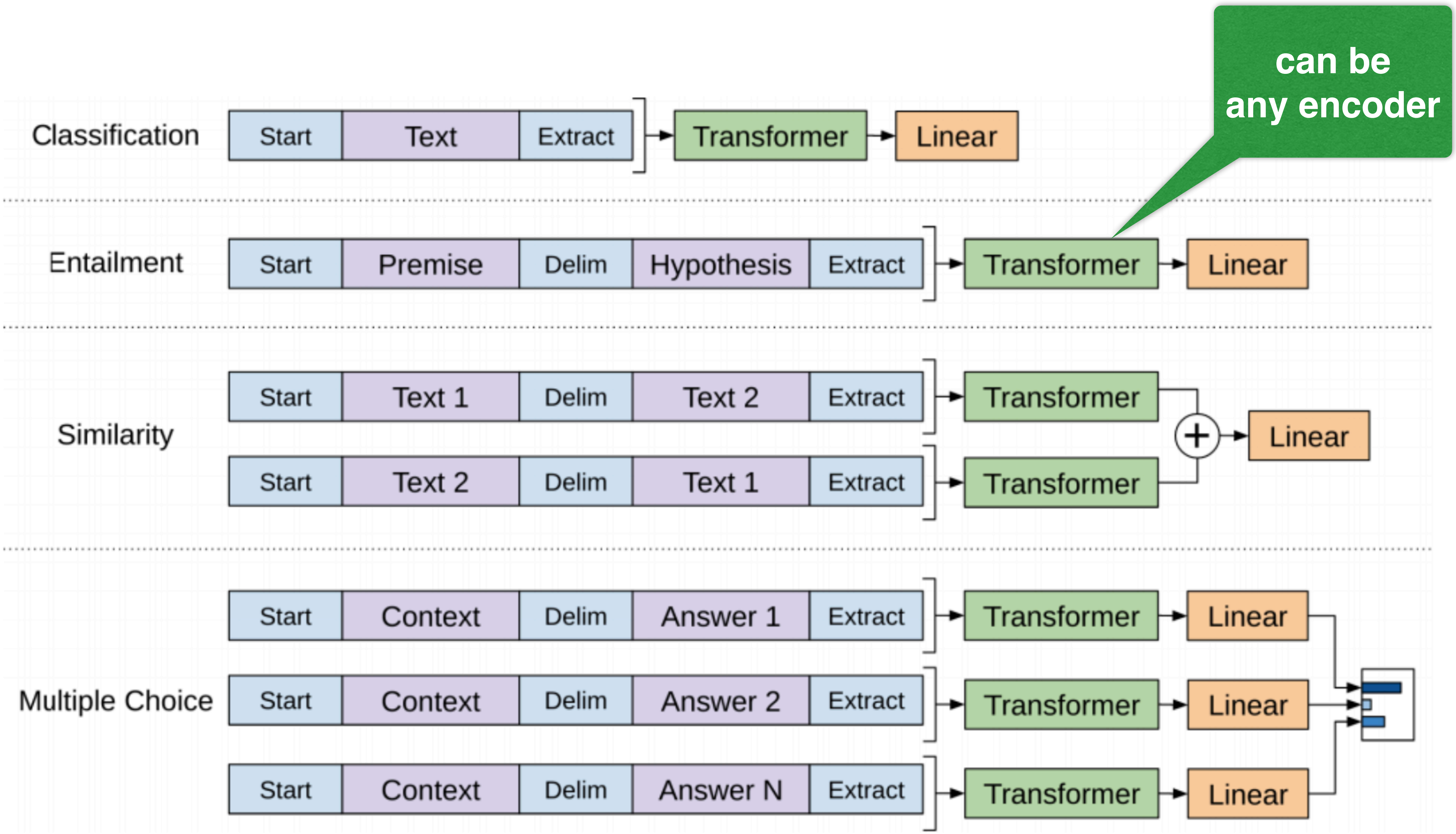


(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Using pre-trained encoders



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

LM Pre-training is very effective.

Further details: papers and blogs.

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

Universal Language Model Fine-tuning for Text Classification

Jeremy Howard*
fast.ai
University of San Francisco
j@fast.ai

Sebastian Ruder*
Insight Centre, NUI Galway
Aylien Ltd., Dublin
sebastian@ruder.io

ULMfit

Deep contextualized word representations

Matthew E. Peters[†], **Mark Neumann**[†], **Mohit Iyyer**[†], **Matt Gardner**[†],
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark*, **Kenton Lee***, **Luke Zettlemoyer**[†*]
{csquared,kentonl,lsz}@cs.washington.edu

[†]Allen Institute for Artificial Intelligence
*Paul G. Allen School of Computer Science & Engineering, University of Washington

ELMo

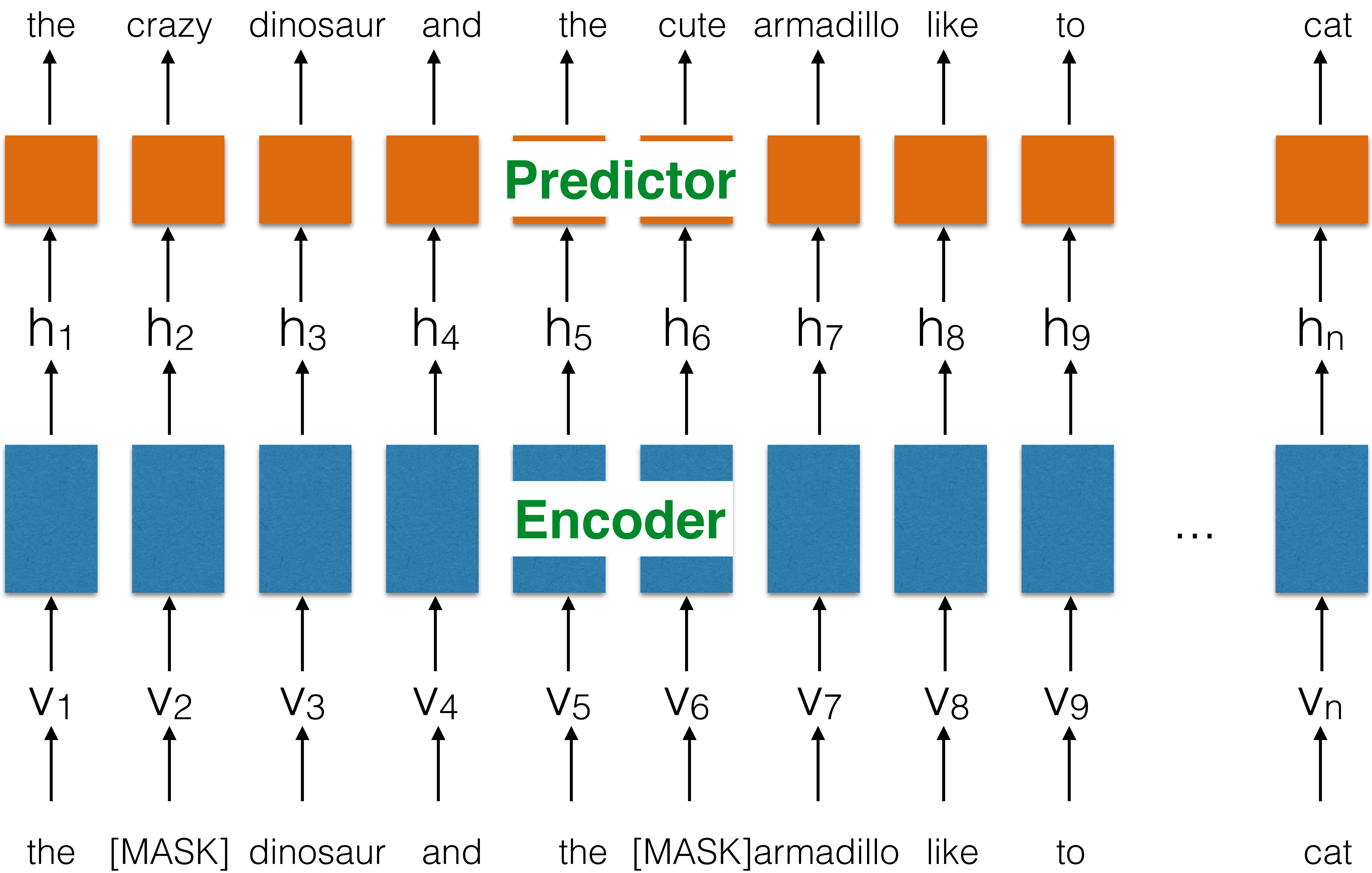
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

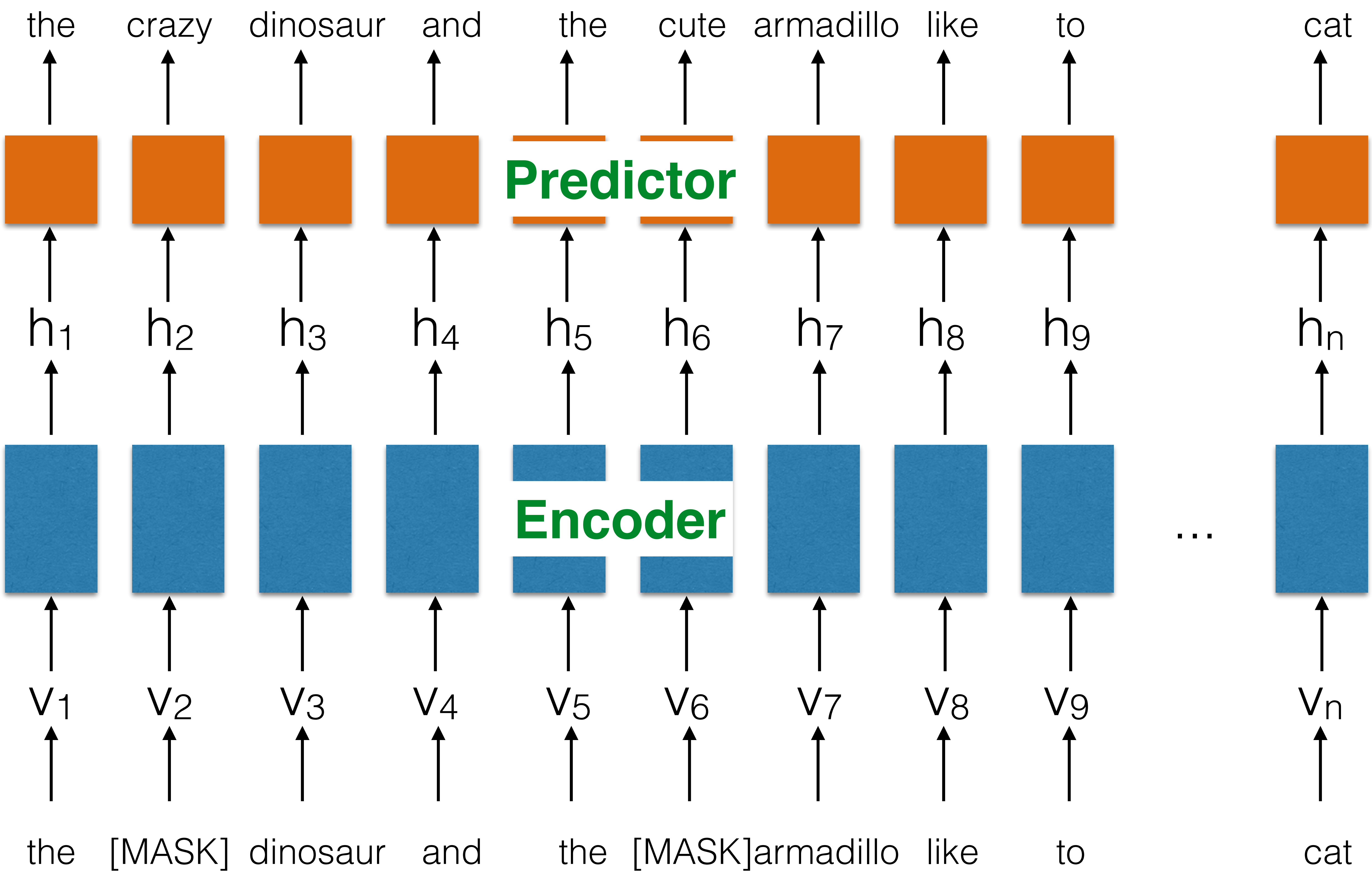
BERT

Some more details:
masked LM, large vocabularies.

at (pre)train time **objectives**

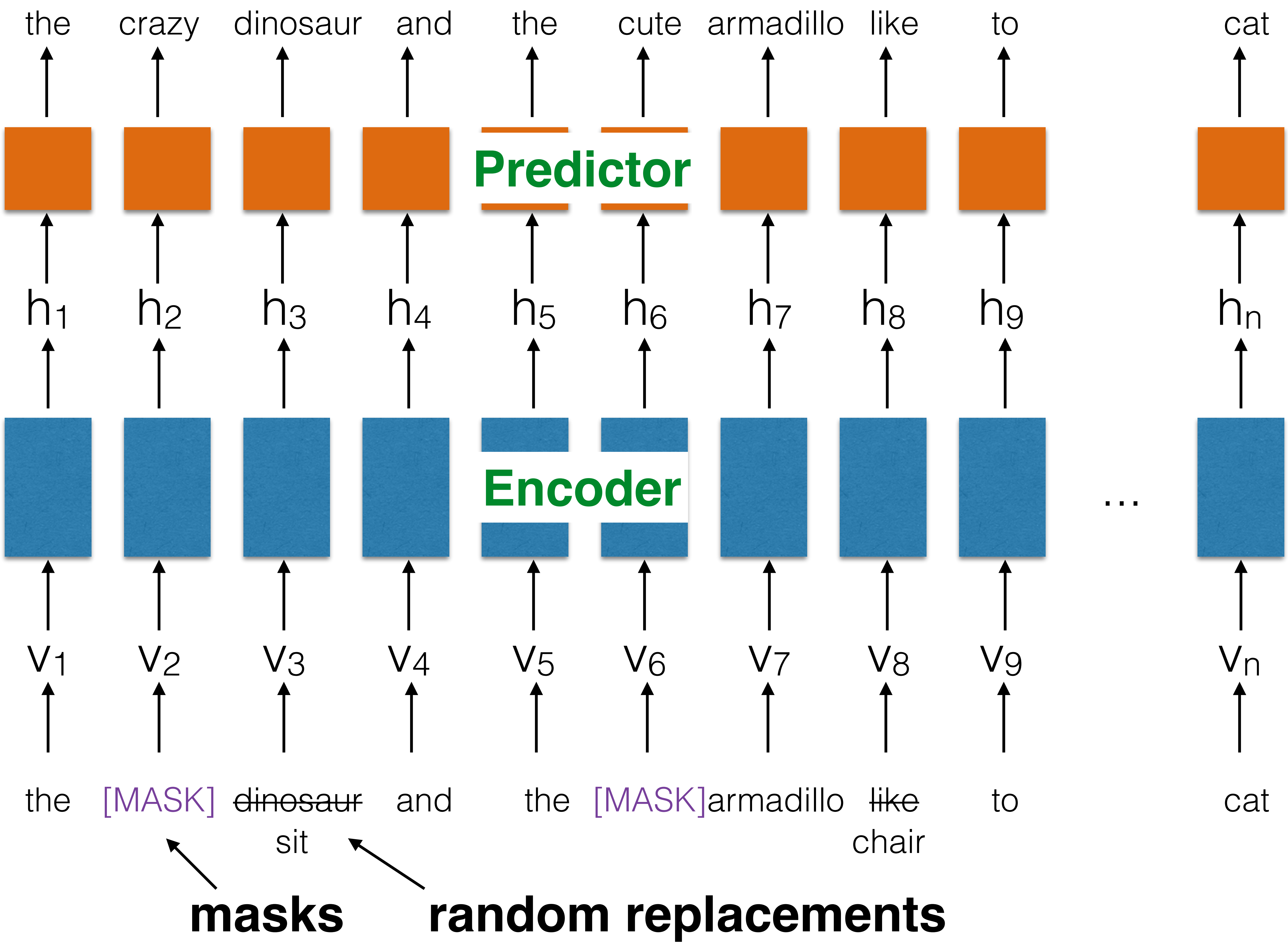


at (pre)train time **objectives**

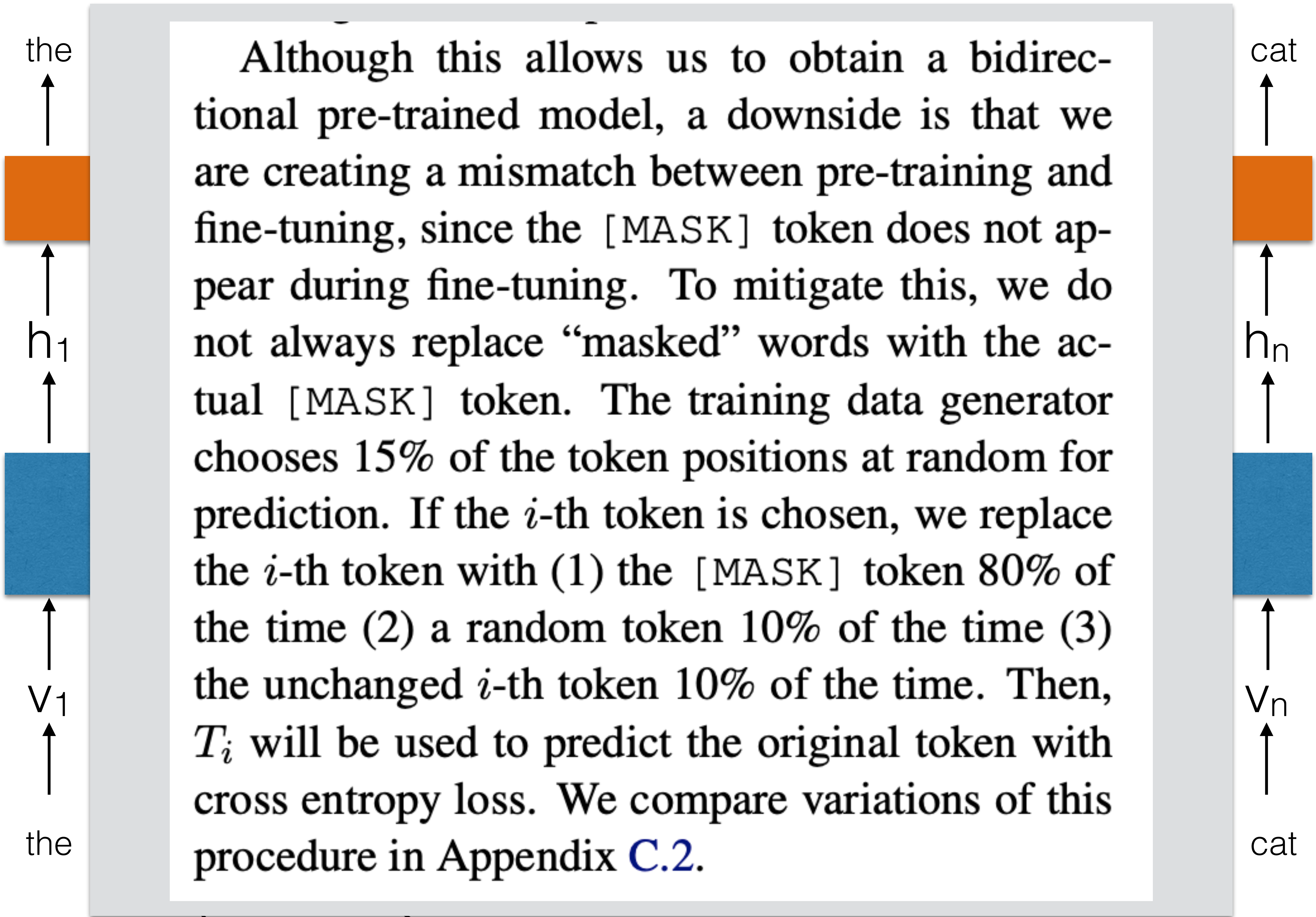


is there a problem here? consider test time.

at (pre)train time **objectives**



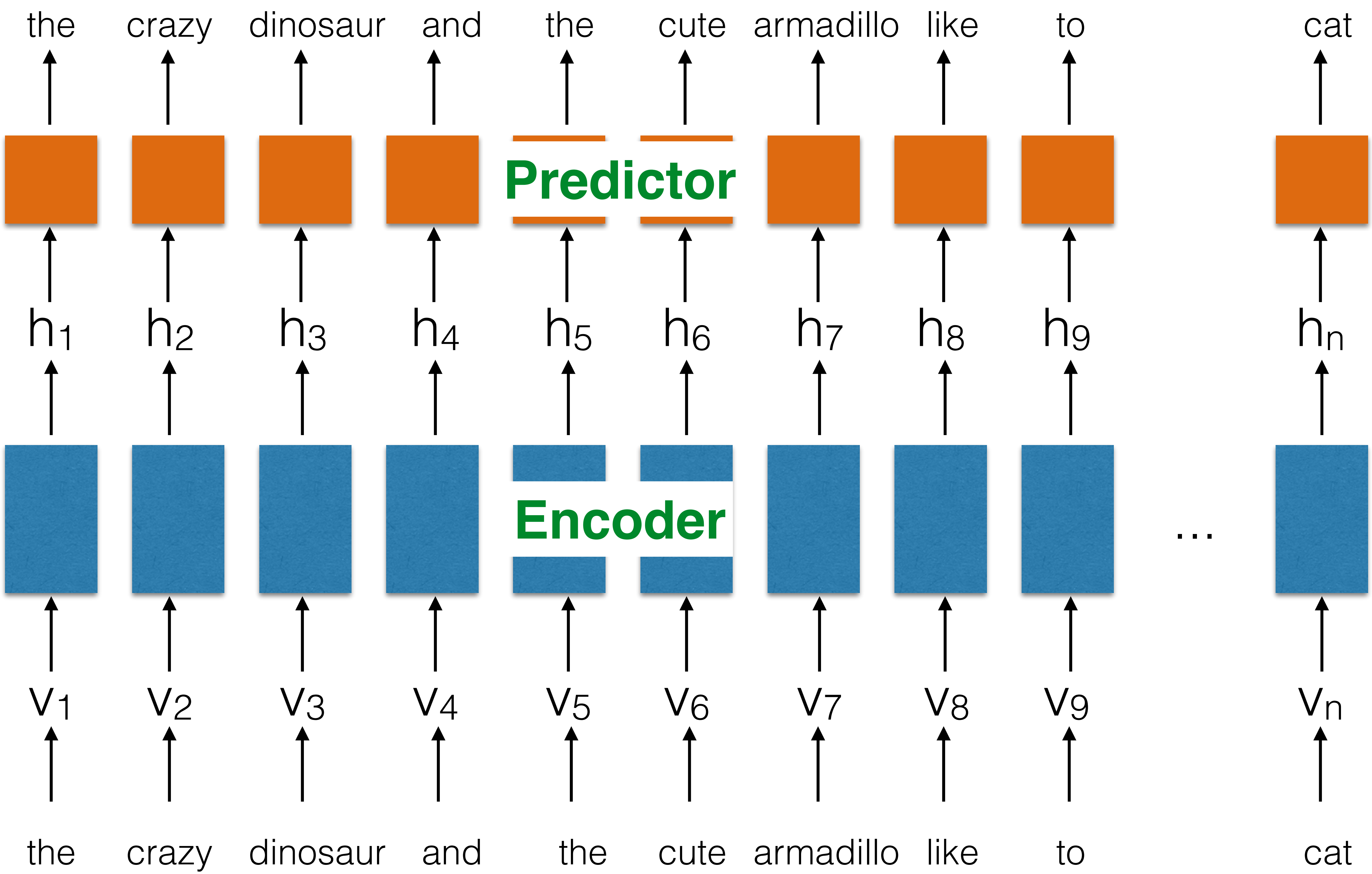
at (pre)train time **objectives**



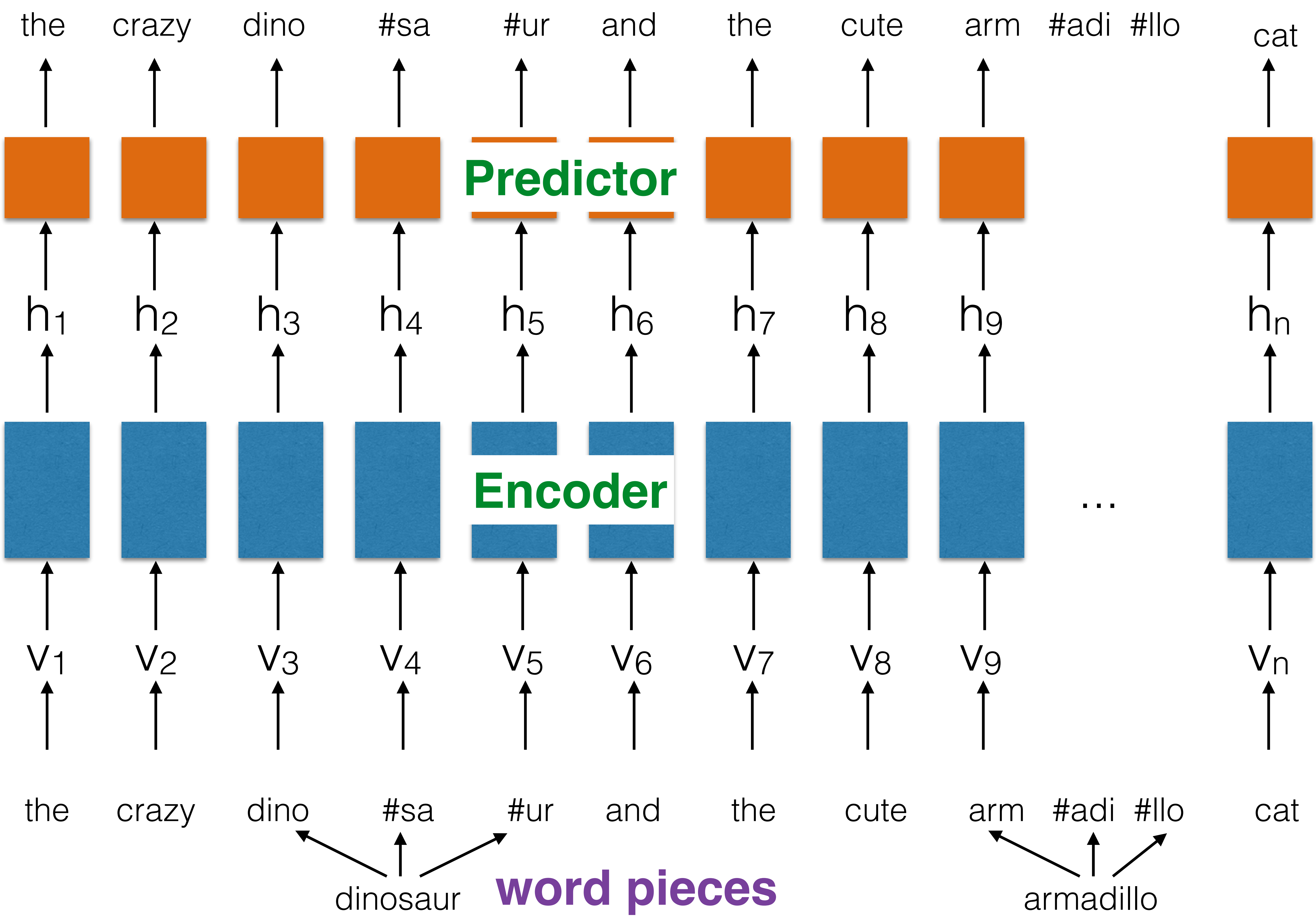
Although this allows us to obtain a bidirectional pre-trained model, a downside is that we are creating a mismatch between pre-training and fine-tuning, since the [MASK] token does not appear during fine-tuning. To mitigate this, we do not always replace "masked" words with the actual [MASK] token. The training data generator chooses 15% of the token positions at random for prediction. If the i-th token is chosen, we replace the i-th token with (1) the [MASK] token 80% of the time (2) a random token 10% of the time (3) the unchanged i-th token 10% of the time. Then, T_i will be used to predict the original token with cross entropy loss. We compare variations of this procedure in Appendix C.2.

masks **random replacements**

dealing with large vocabulary and unknown words

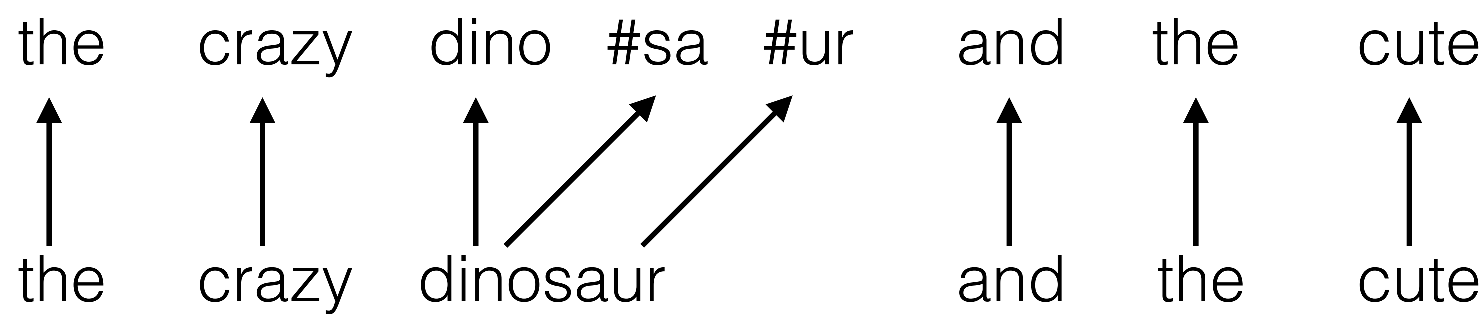


dealing with large vocabulary and unknown words



dealing with large vocabulary and unknown words

word pieces



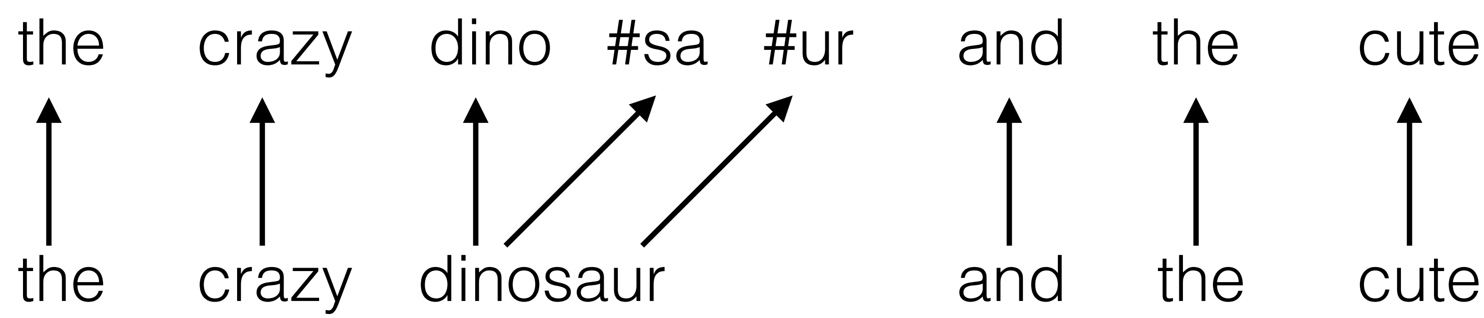
Reduce the vocabulary by (deterministically) cutting some symbols into smaller pieces.

In the extreme case --> just use characters as inputs.
(In the more extreme case --> just use bytes) **"tokenizer free"**

We seek a middle ground: capture frequent larger units.
Allow a "budget" of k vocabulary items, choose basic units to fill this space. (typical k: 30,000. why? GPU constraints.)

dealing with large vocabulary and unknown words

word pieces



Allow a "budget" of k vocabulary items, choose basic units to fill this space.

Algorithms:

- "Word pieces" (at Google)
- **BPE (rest of the world)**

Neural Machine Translation of Rare Words with Subword Units

Rico Sennrich and **Barry Haddow** and **Alexandra Birch**
School of Informatics, University of Edinburgh
{rico.sennrich,a.birch}@ed.ac.uk,bhaddow@inf.ed.ac.uk

dealing with large vocabulary and unknown words

word pieces

r ·	→	r·
l o	→	lo
lo w	→	low
e r·	→	er·

Figure 1: BPE merge operations learned from dictionary {'low', 'lowest', 'newer', 'wider'}.

Algorithms:

- "Word pieces" (at Google)
- **BPE (rest of the world)**

Firstly, we initialize the symbol vocabulary with the character vocabulary, and represent each word as a sequence of characters, plus a special end-of-word symbol '·', which allows us to restore the original tokenization after translation. We iteratively count all symbol pairs and replace each occurrence of the most frequent pair ('A', 'B') with a new symbol 'AB'. Each merge operation produces a new symbol which represents a character n-gram. Frequent character n-grams (or whole words) are eventually merged into a single symbol, thus BPE requires no shortlist. The final symbol vocabulary size is equal to the size of the initial vocabulary, plus the number of merge operations – the latter is the only hyperparameter of the algorithm.

Neural Machine Translation of Rare Words with Subword Units

Rico Sennrich and **Barry Haddow** and **Alexandra Birch**
School of Informatics, University of Edinburgh
{rico.sennrich,a.birch}@ed.ac.uk, bhaddow@inf.ed.ac.uk

BERT Recap

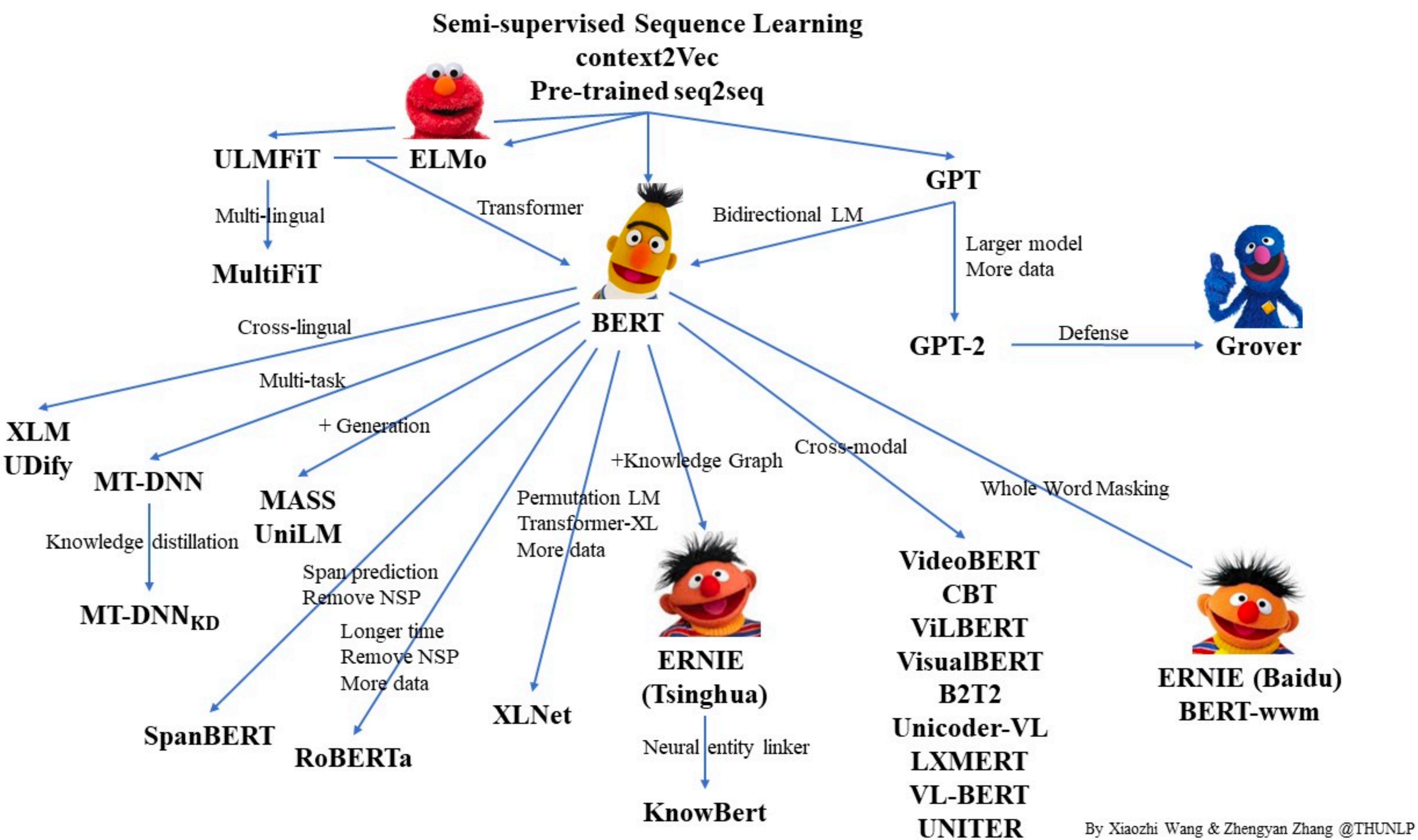
- Core idea of BERT: "**masked language model"**
    - Another view on this task / idea:
      "**sequence denoising**"
      "**denoising autoencoder**".
- Also in BERT: **"next sentence prediction"**.

  ("are these two sentences compatible or not?")

(What does it remind you of?)

BERT Variants

BERT Variants



BERT Variants

BERT-base   BERT-large   ....

BERT Variants

BERT-base BERT-large ....

BERT-WWM

(whole-word-masking)

BERT Variants

RoBERTA

BERT Variants

RoBERTA

Our modifications are simple, they include: (1) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences; and (4) dynamically changing the masking pattern applied to the training data. We also collect a large new dataset (CC-NEWS) of comparable size to other privately used datasets, to better control for training set size effects.

BERT Variants

RoBERTA

Our modifications are simple, they include: (1) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences; and (4) dynamically changing the masking pattern applied to the training data. We also collect a large new dataset (CC-NEWS) of comparable size to other privately used datasets, to better control for training set size effects.

BERT Variants

RoBERTA

4.1 Static vs. Dynamic Masking

As discussed in Section 2, BERT relies on randomly masking and predicting tokens. The original BERT implementation performed masking once during data preprocessing, resulting in a single *static* mask. To avoid using the same mask for each training instance in every epoch, training data was duplicated 10 times so that each sequence is masked in 10 different ways over the 40 epochs of training. Thus, each training sequence was seen with the same mask four times during training.

We compare this strategy with *dynamic masking* where we generate the masking pattern every time we feed a sequence to the model. This becomes crucial when pretraining for more steps or with larger datasets.

BERT Variants

RoBERTA

- Train for longer.
- Train on more data.
- "Do the right thing" with the masking.
  (b/c TF vs PyTorch? technology tools matter!)

BERT Variants

RoBERTA

- Train for longer.
- Train on more data.
- "Do the right thing" with the masking.
  (b/c TF vs PyTorch? technology tools matter!)

RoBERTA performs **much** better
than BERT on many cases

BERT Variants

oLMpics - On what Language Model Pre-training Captures

Alon Talmor[1,2] **Yanai Elazar**[1,3] **Yoav Goldberg**[1,3] **Jonathan Berant**[1,2]
[1]The Allen Institute for AI
[2]Tel-Aviv University
[3]Bar-Ilan University
{alontalmor@mail,joberant@cs}.tau.ac.il
{yanaiela,yoav.goldberg}@gmail.com

RoBERTA performs **much** better than BERT on many cases

BERT Variants

SpanBERT

- No NSP (like RoBERTa)
- Mask whole spans. Predict each word from boundary + relative position.

BERT Variants

SpanBERT

- No NSP (like RoBERTa)
- Mask whole spans. Predict each word from boundary + relative position.

$$\begin{aligned}\mathcal{L}(\text{football}) &= \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football}) \\ &= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)\end{aligned}$$

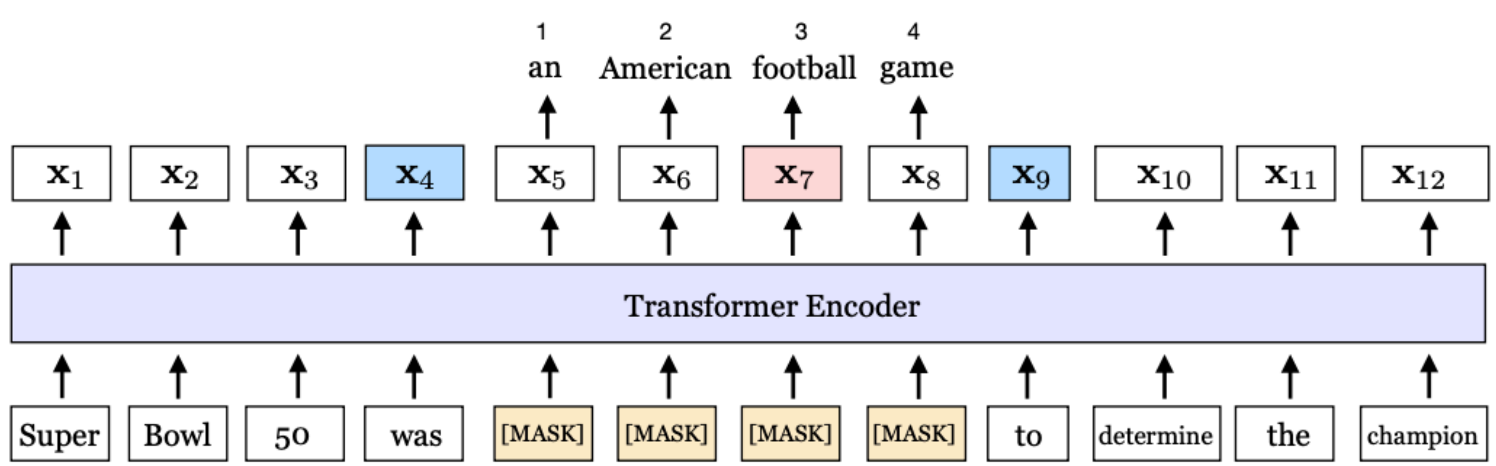


Figure 1: An illustration of SpanBERT training. The span *an American football game* is masked. The span boundary objective (SBO) uses the output representations of the boundary tokens, $\mathbf{x}_4$ and $\mathbf{x}_9$ (in blue), to predict each token in the masked span. The equation shows the MLM and SBO loss terms for predicting the token, *football* (in pink), which as marked by the position embedding $\mathbf{p}_3$, is the *third* token from x_4.

BERT Variants

ALBERT

- Main idea: larger models with same memory / same parameters count
- Replace NSP with SOP (sentence order prediction)

BERT Variants

ALBERT

Param-count reduction:

(1) Factorizing the embedding matrices: $W = U_1 U_2$

Instead of one V x H matrix:

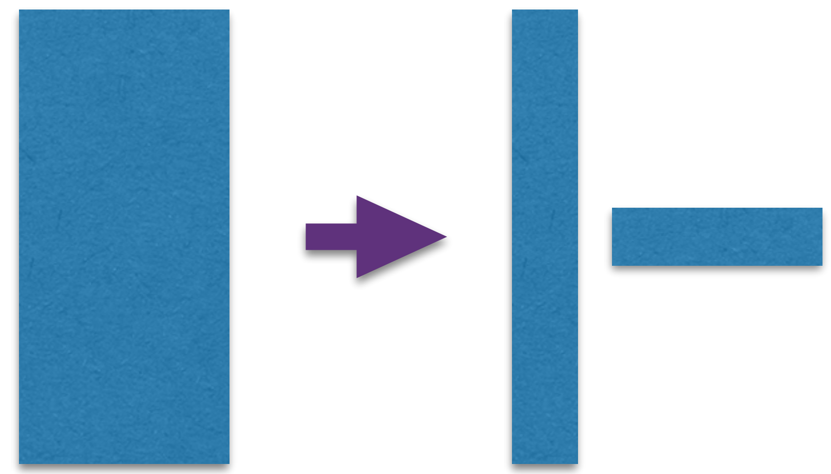
one V x E matrix, one E x H matrix

Do this for both embedding matrices

(what is the maximal rank of the ALBERT embedding matrix vs the regular one?)

BERT Variants

ALBERT



Param-count reduction:

(1) Factorizing the embedding matrices: $W = U_1 U_2$

Instead of one V x H matrix:

one V x E matrix, one E x H matrix

Do this for both embedding matrices

(what is the maximal rank of the ALBERT embedding matrix vs the regular one?)

BERT Variants

ALBERT

Param-count reduction:

(2) Parameter sharing across layers

(what it sounds like)

BERT Variants

ALBERT

NSP -> SOP

The NSP task is too simple for the model.

NSP can be "solved" by learning "topical match" between the sentences, not necessarily order or deep semantics.

Sentence order prediction: show two sentences, either in the right order or in reverse order. Model needs to which is which.

SOP is much stronger than NSP

BERT Variants

ALBERT

NSP -> SOP

Param-count reduction:

(1) Factorizing the embedding matrices: $W = U_1 U_2$

(2) Parameter sharing across layers

BERT Variants

ELECTRA

Generative --> Discriminative

Instead of masking the input, our approach corrupts it by replacing some tokens with plausible alternatives sampled from a small generator network. Then, instead of training a model that predicts the original identities of the corrupted tokens, we train a discriminative model that predicts whether each token in the corrupted input was replaced by a generator sample or not. Thorough experiments demonstrate this new pre-training task is more efficient than MLM because the task is defined over *all* input tokens rather than just the small subset that was masked out. As a result, the contextual representations learned by our approach substantially outperform the ones learned by BERT given the same model size, data, and compute.

BERT Variants

RoBERTA

SpanBERT

ALBERT

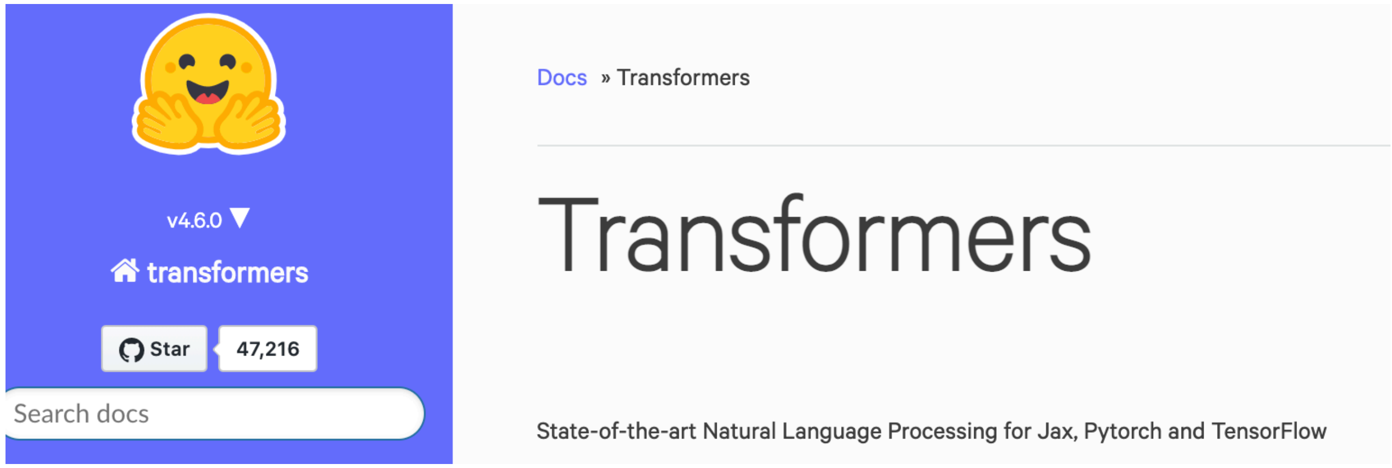
ELECTRA

Canine <-- char-level model

DeBERTa <-- current "best"

XLM, XLNet, ...

Many others



BERT Variants

RoBERTA

SpanBERT

ALBERT

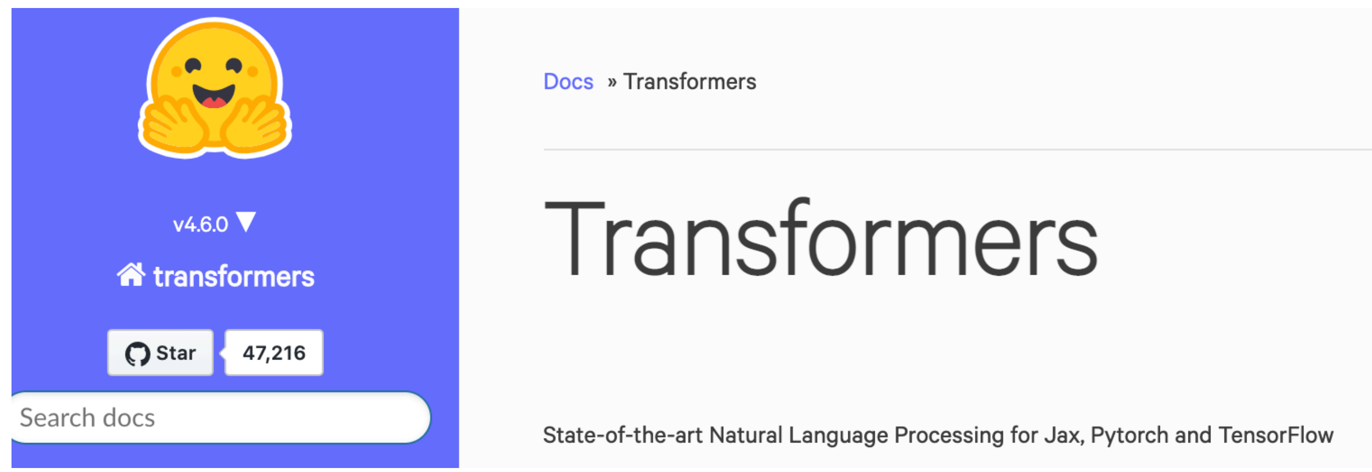
ELECTRA

Canine

DeBERTa

XLM, XLNet, ...

Many others



domains --> **BERTweet** **SciBERT**

languages --> **CamemBERT** **AlephBERT**

multilingual --> **mBERT**

BERT Variants

- All encode **n input tokens** into **n output vectors**.
- All share the same main **"sequence denoising" objective.**

(what are the differences? why do they matter?
can you think of additional variants?)

Beyond GPT and BERT

Generative models / Seq-seq

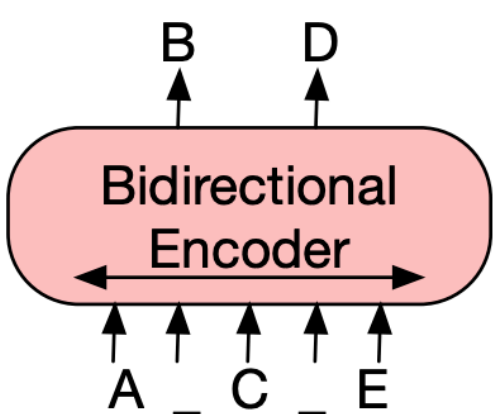
T5 **BART**

Beyond GPT and BERT

Generative models / Seq-seq

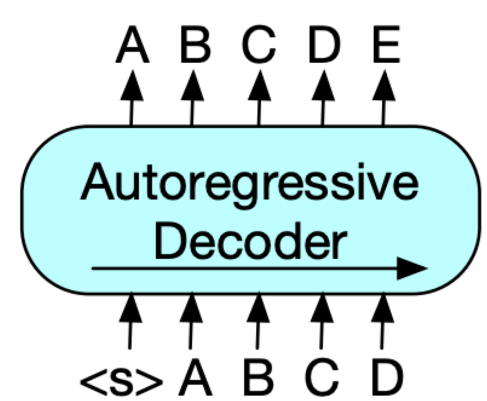
previously seen models:

BERT/MLM



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

GPT/LM



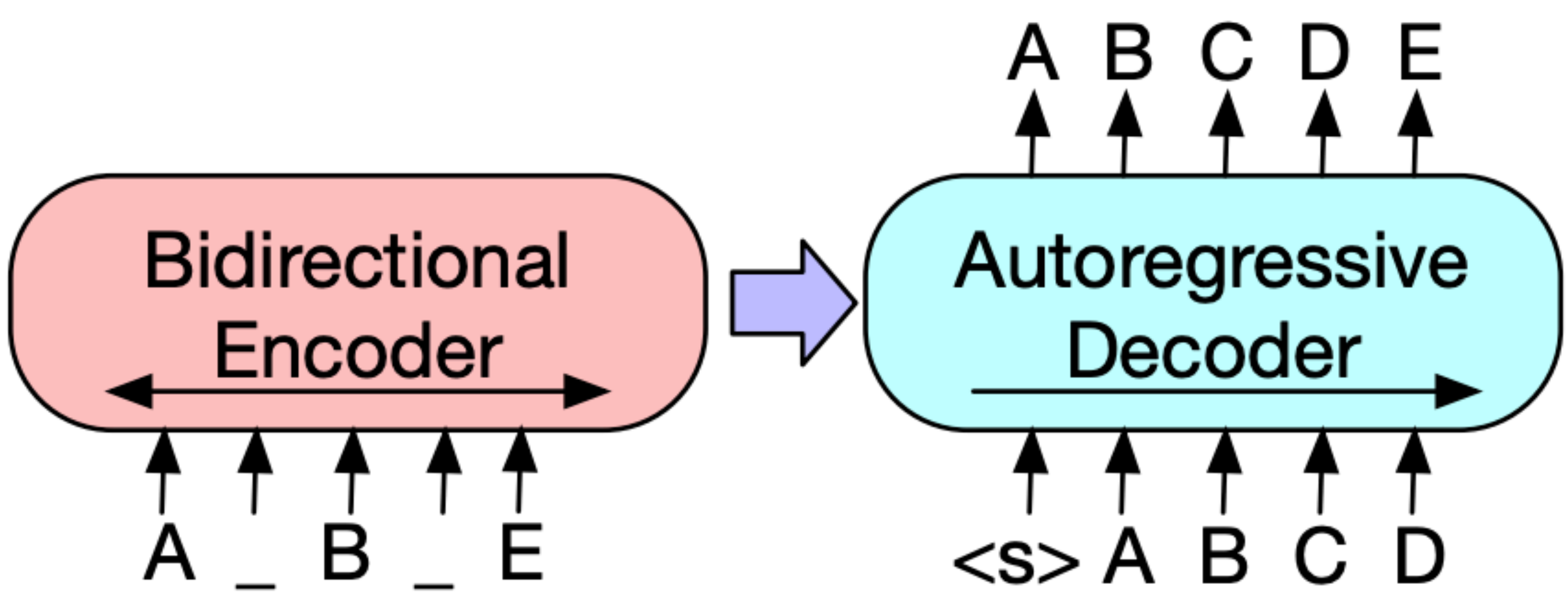
(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

Images from BART paper

Beyond GPT and BERT

Generative models / Seq-seq

T5 & BART



Images from BART paper

T5:
Text to Text Transfer Transformer

- **Retain the "denoising" / "cloze completion" objective.**

- **Perform seq2seq (encode->decode) instead of MLM**

T5:
Text to Text Transfer Transformer

- **Retain the "denoising" / "cloze completion" objective.**

- **Perform seq2seq (encode->decode) instead of MLM**

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

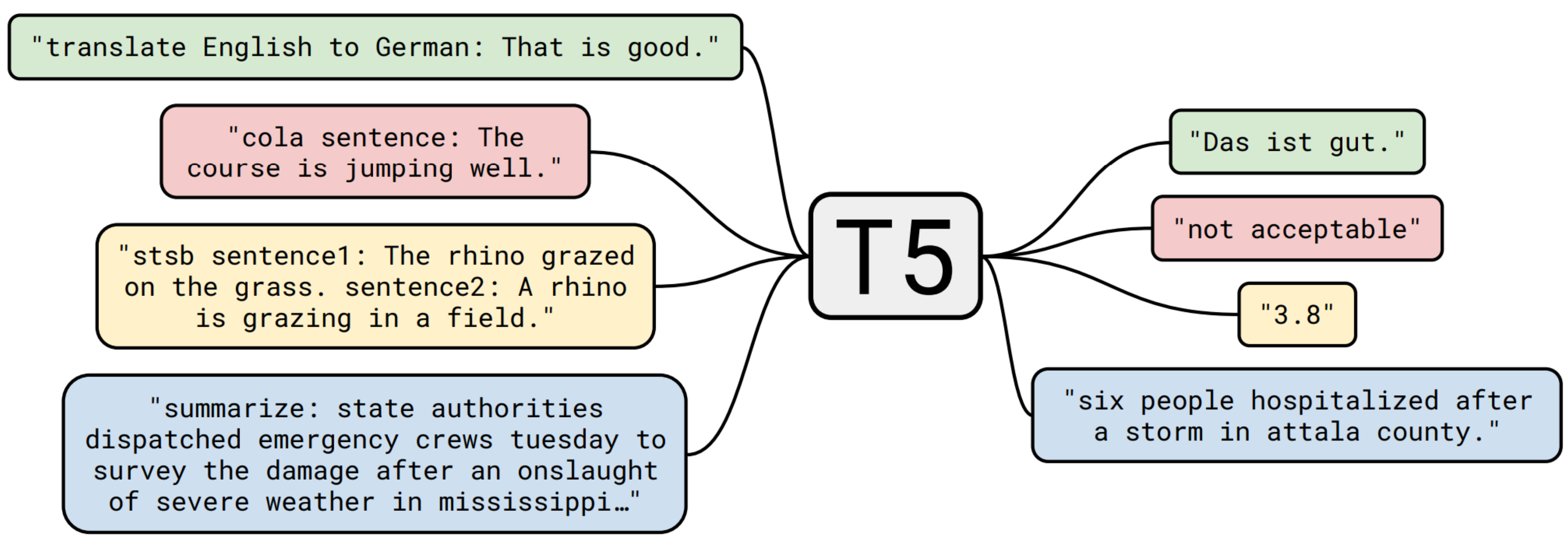
Colin Raffel* CRAFFEL@GMAIL.COM
Noam Shazeer* NOAM@GOOGLE.COM
Adam Roberts* ADAROB@GOOGLE.COM
Katherine Lee* KATHERINELEE@GOOGLE.COM
Sharan Narang SHARANNARANG@GOOGLE.COM
Michael Matena MMATENA@GOOGLE.COM
Yanqi Zhou YANQIZ@GOOGLE.COM
Wei Li MWEILI@GOOGLE.COM
Peter J. Liu PETERJLIU@GOOGLE.COM

Google, Mountain View, CA 94043, USA

T5:
Text to Text Transfer Transformer

- Very large model.

- Train many supervised text-to-text models jointly.



- Train on very large training data (how?)

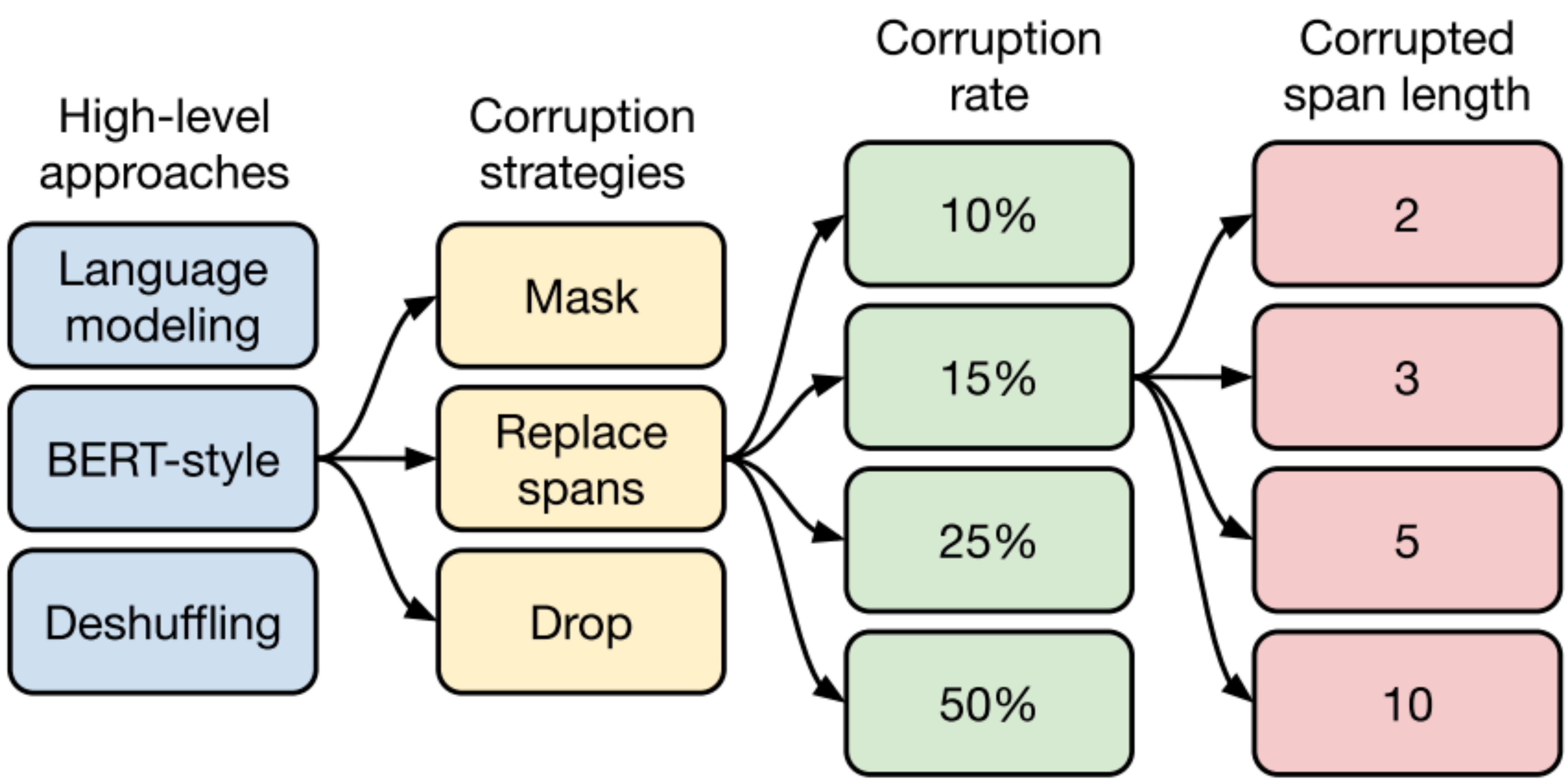
Images from T5 paper

T5:
Text to Text Transfer Transformer

- de-masking / denoising as Text-to-text:

T5:
Text to Text Transfer Transformer

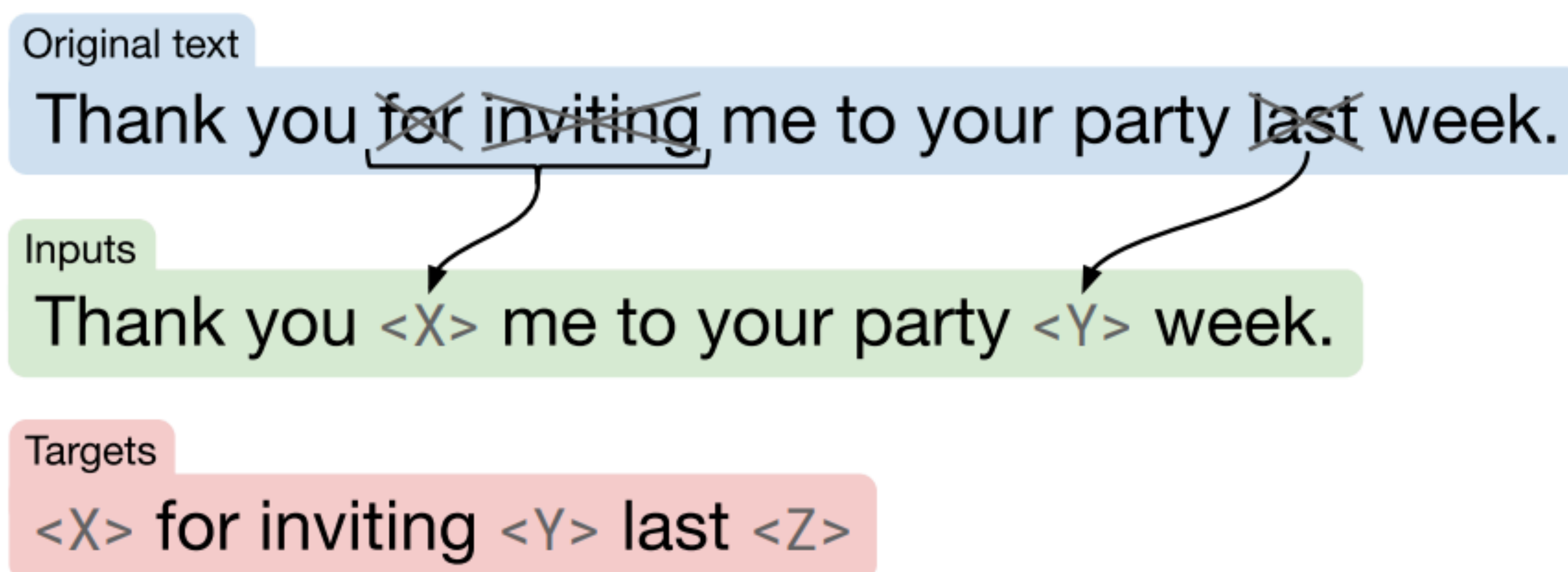
- de-masking / denoising as Text-to-text:
  - Extensive set of experiments:



Images from T5 paper

T5:
Text to Text Transfer Transformer

- de-masking / denoising as Text-to-text:



Images from T5 paper

T5:
Text to Text Transfer Transformer

- de-masking / denoising as Text-to-text:

Alternatives:

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

T5:
Text to Text Transfer Transformer

- ByT5 --> same thing but on bytes and not subwords
- mT5   --> same things but multilingual

BART

- various de-noising objectives

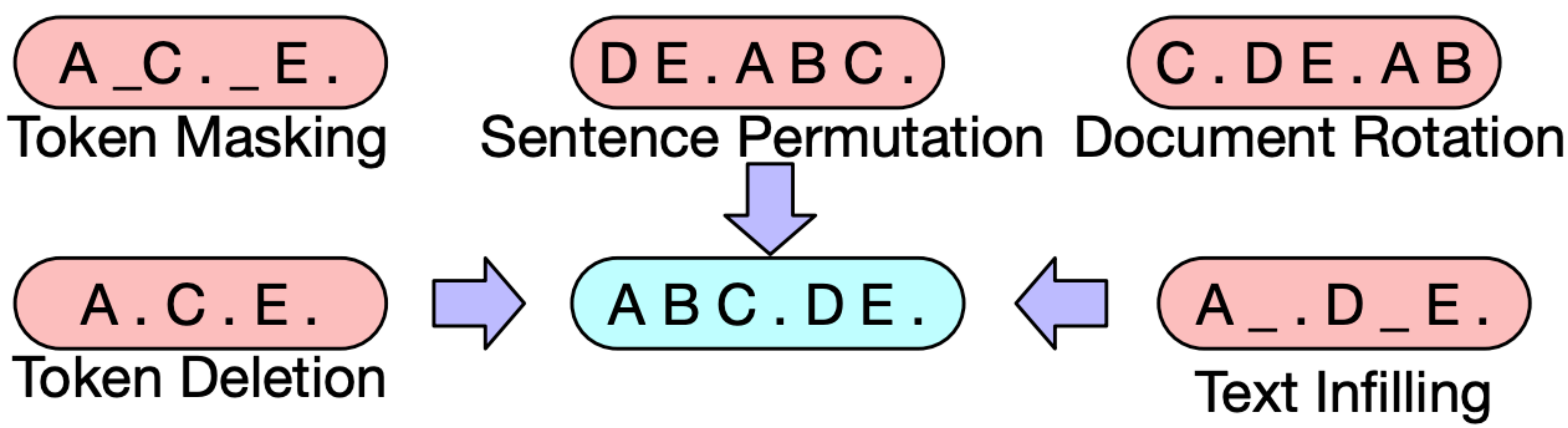
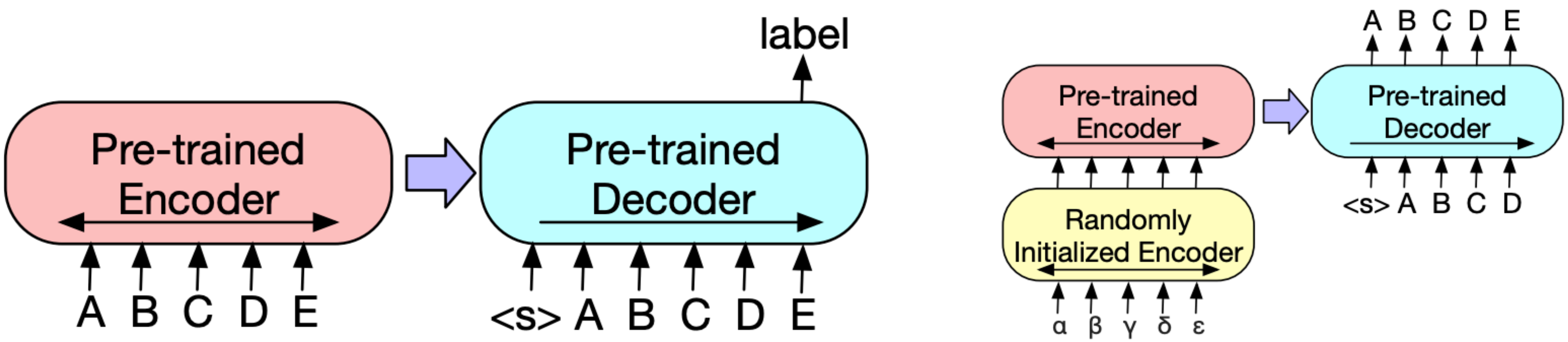


Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

Images from BART paper

Using BART / T5



(a) To use BART for classification problems, the same input is fed into the encoder and decoder, and the representation from the final output is used.

(b) For machine translation, we learn a small additional encoder that replaces the word embeddings in BART. The new encoder can use a disjoint vocabulary.

Figure 3: Fine tuning BART for classification and translation.

Images from BART paper

Using BART / T5

- And of course, you can also sample from them!

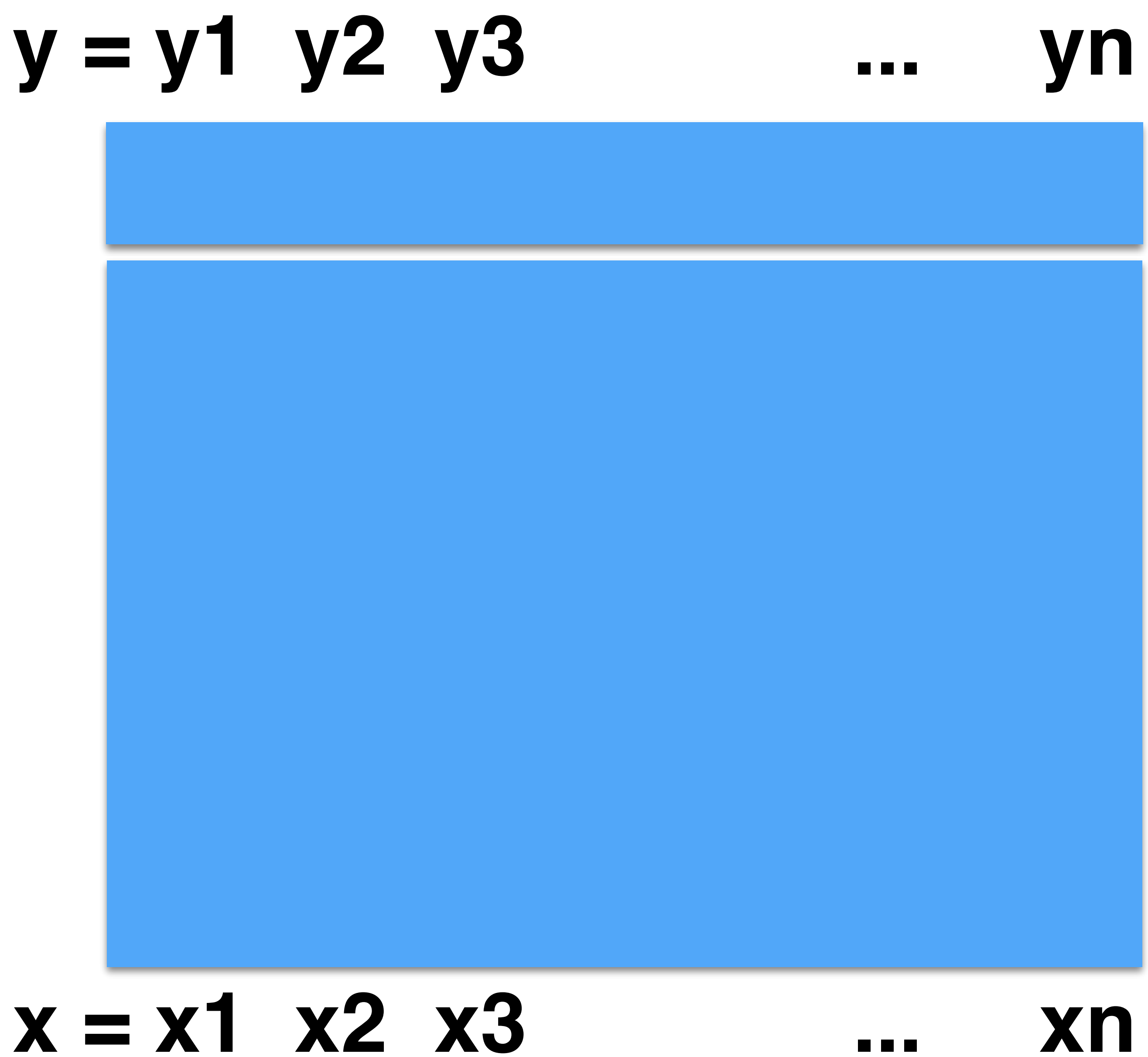
Recap pre-trained LMs

- Train an LM, n-to-n encoder, or n-to-m encoder-decoder with a denoising LM objective over large corpus for a long time.
- Drop the prediction layer.
- Get effective general purpose encoder, that can be easily "fine-tuned" to other tasks.
- (why does it work?)

ULmfit, GPT, GPT2,3,...	Elmo/BERT	T5/BART
Language Model	Bidi-LM	Encode-Decode
n to 1 **or** 0 to n **(depending on your p.o.v)**	n to n	n to m

Using a Pre-trained Network

an abstract network structure: encoder + predictor

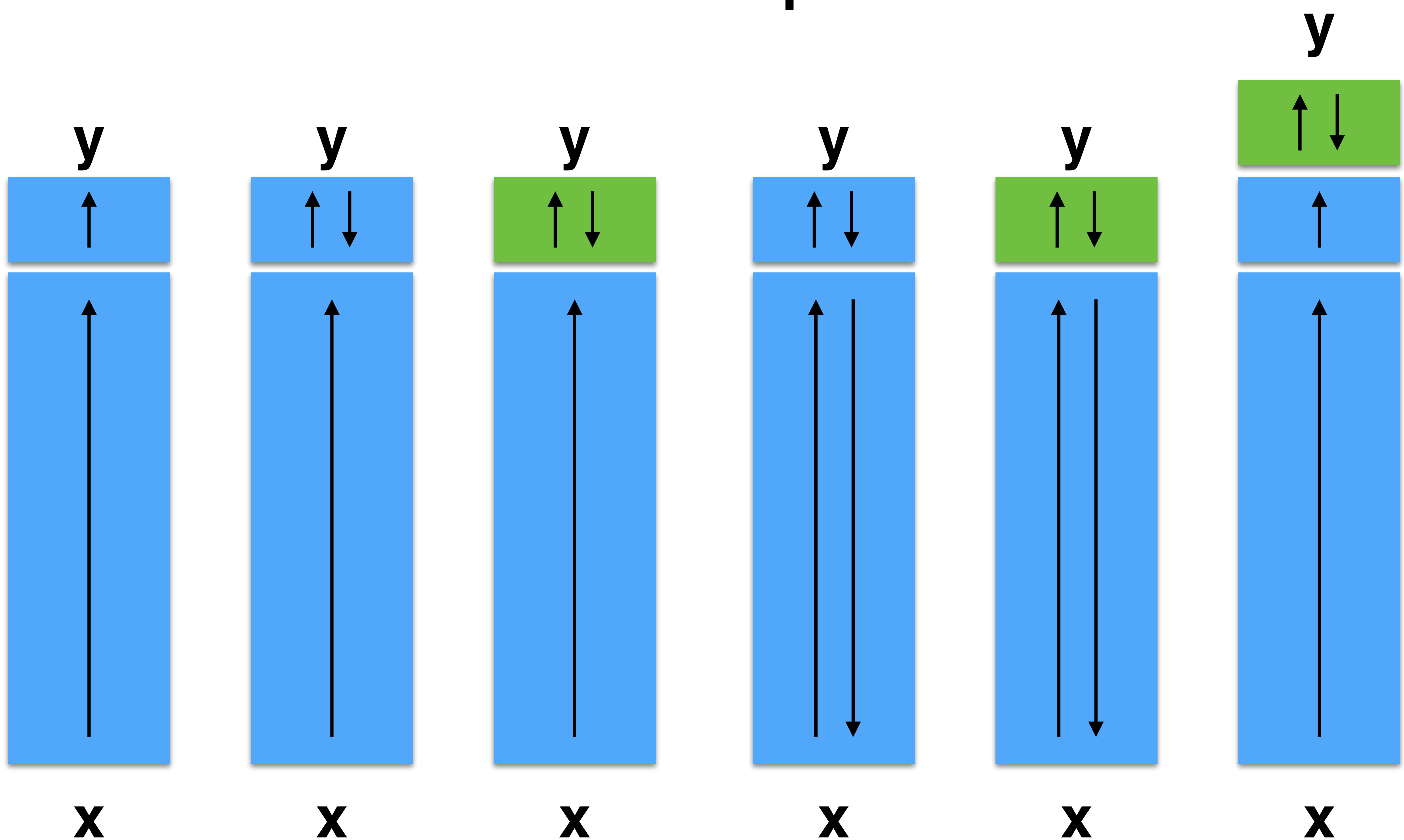


Using a Pre-trained Network

a more concise drawing

Using a Pre-trained Network

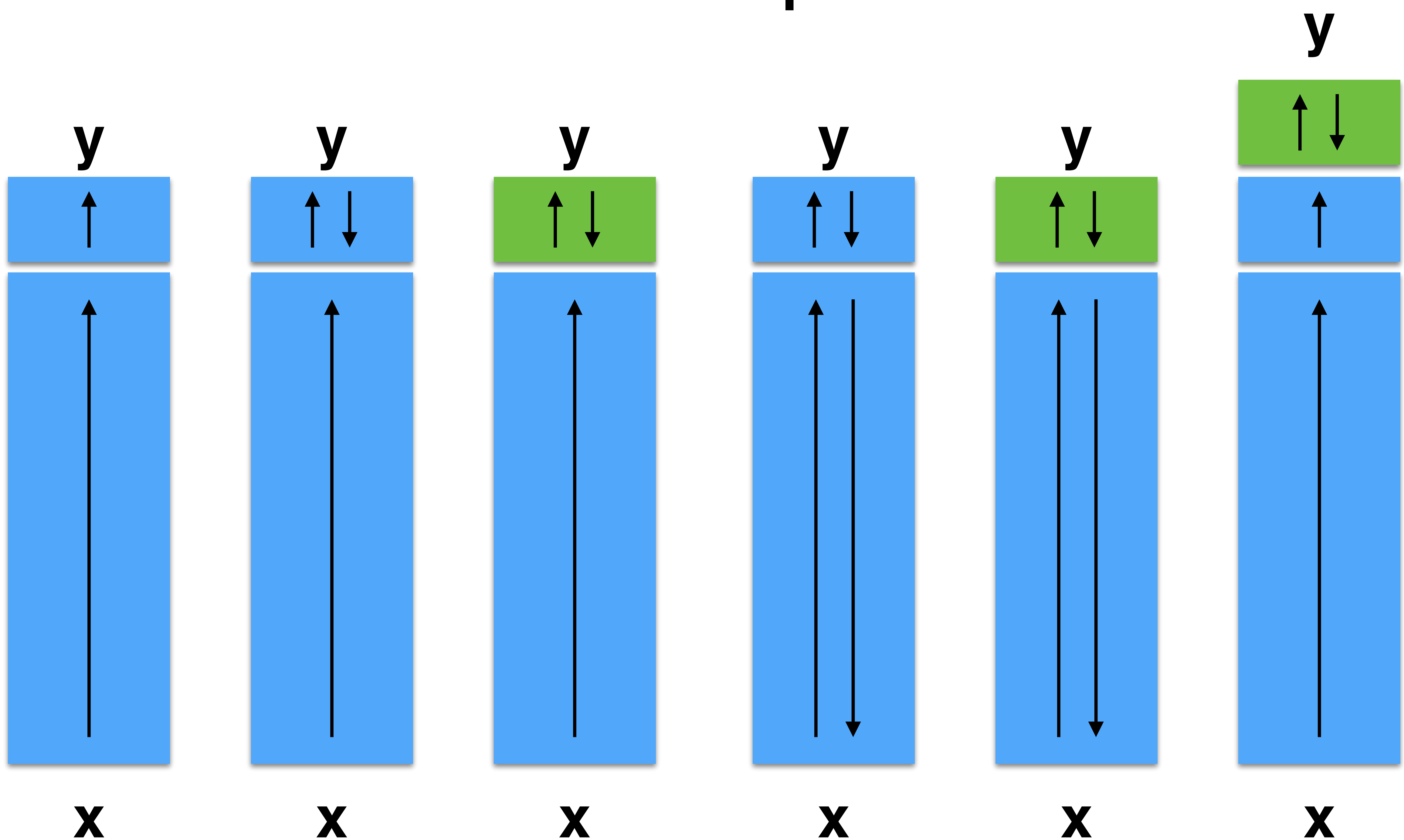
the different options



fixed: ↑ tuned: ↑ ↓ new component existing component

Using a Pre-trained Network

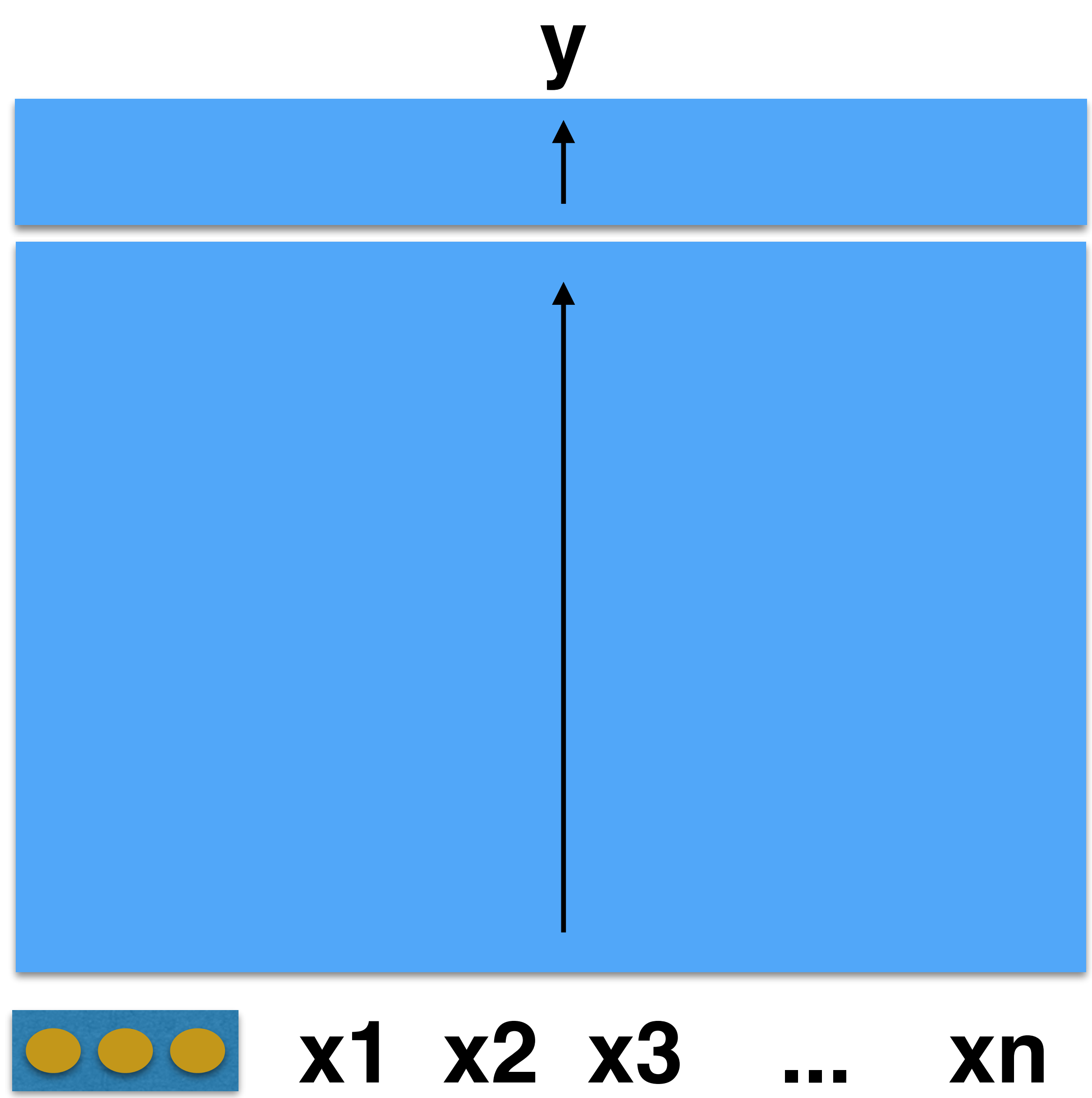
the different options



Using a Pre-trained Network

another option ("prompt tuning")

add vectors
to the input
and train them.



BERT-ology

- What is captured by a pre-trained masked LM?
- What things are learned at different layer?
- What can it learn? what can't it learn?
- many questions, active research field.

BERT-ology

A Primer in BERTology: What We Know About How BERT Works

Anna Rogers
Center for Social Data Science
University of Copenhagen
arogers@sodas.ku.dk

Olga Kovaleva
Dept. of Computer Science
University of Massachusetts Lowell
okovalev@cs.uml.edu

Anna Rumshisky
Dept. of Computer Science
University of Massachusetts Lowell
arum@cs.uml.edu

Cost / "green AI"

Energy and Policy Considerations for Deep Learning in NLP

Emma Strubell **Ananya Ganesh** **Andrew McCallum**
College of Information and Computer Sciences
University of Massachusetts Amherst
{strubell, aganesh, mccallum}@cs.umass.edu

Cost / "green AI"

Model	Hardware	Power (W)	Hours	kWh·PUE	CO_2e	Cloud compute cost
$\text{Transformer}_{base}$	P100x8	1415.78	12	27	26	\$41–\$140
Transformer_{big}	P100x8	1515.43	84	201	192	\$289–\$981
ELMo	P100x3	517.66	336	275	262	\$433–\$1472
BERT_{base}	V100x64	12,041.51	79	1507	1438	\$3751–\$12,571
BERT_{base}	TPUv2x16	—	96	—	—	\$2074–\$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	\$942,973–\$3,201,722
NAS	TPUv2x1	—	32,623	—	—	\$44,055–\$146,848
GPT-2	TPUv3x32	—	168	—	—	\$12,902–\$43,008

Table 3: Estimated cost of training a model in terms of CO_2 emissions (lbs) and cloud compute cost (USD).[7] Power and carbon footprint are omitted for TPUs due to lack of public information on power draw for this hardware.

Cost / "green AI"

Consumption	CO_2e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Cost / "green AI"

how do we train smaller / more efficient models?

how do we train models that can learn from fewer data?

Distillation

Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†]
Google Inc.
Mountain View
geoffhinton@google.com

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

Jeff Dean
Google Inc.
Mountain View
jeff@google.com

- Train a smaller network on the output of a larger network.
- The smaller network mimics the entire output vector, not just the argmax prediction.
- Smaller network can find good solutions, getting close to the larger one.

"DistilBERT"

Compression

- Replacing matrices with smaller matrices
- Replacing 32bit floating point with smaller numbers
    - 8bit
    - 3bit
    - 1bit

Pruning

- Can we identify parameters that can be removed?
- Can we identify blocks that can we removed?
- Can identify sub-networks that can be removed?

Lottery Ticket Hypothesis

- Only some of the parameters are important.
- Popular and easy to read paper. Beyond our scope. Read it.

THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

When BERT Plays the Lottery, All Tickets Are Winning

Sai Prasanna*
Zoho Labs
Zoho Corporation
Chennai, India
saiprasanna.r@zohocorp.com

Anna Rogers*
Center for Social Data Science
Copenhagen University
Copenhagen, Denmark
arogers@sodas.ku.dk

Anna Rumshisky
Dept. of Computer Science
Univ. of Massachusetts Lowell
Lowell, USA
arum@cs.uml.edu

Efficient Fine-tuning

- Can we avoid fine-tuning entire network for each task?
  - Prompt fine-tuning
  - Adapters
  - BitFit <--- tune only the bias parameters

$$Wx + b$$

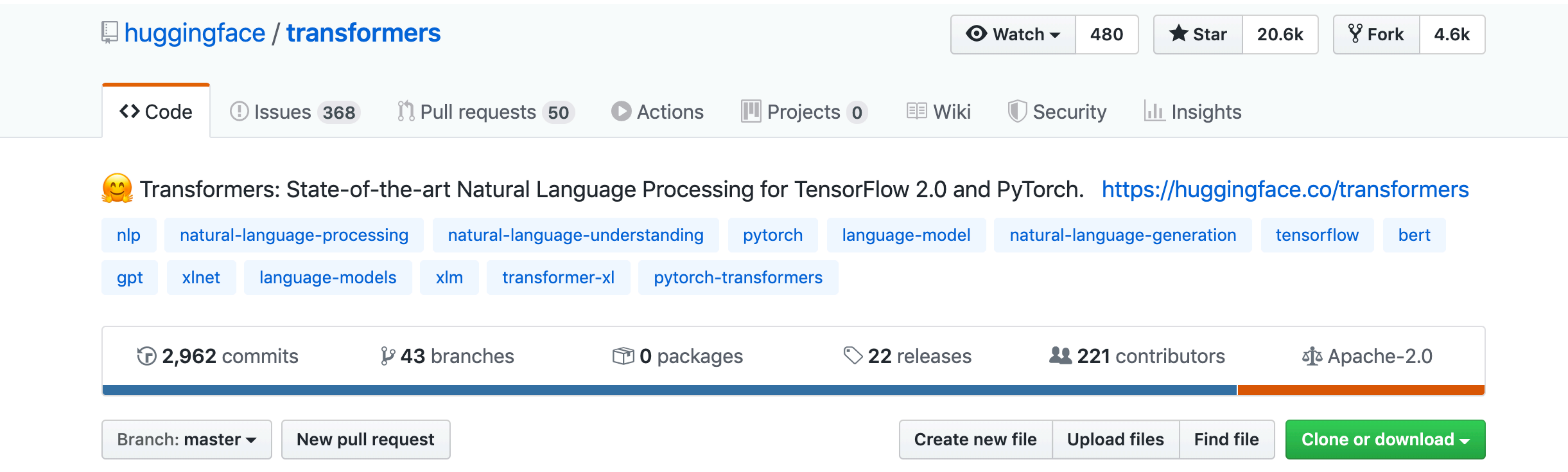
Re-use?

- Can we somehow re-use computation?
- How?
    - Many potential places for re-use, but the details are still an open problem.

Multilingual models

- Train same models on multiple languages
- Impressive cross-lingual transfer results
  - but how / why?

Software



pre-trained encoders, easy to use library.

Vector space alignment

Motivation - Mikolov et al. 2013

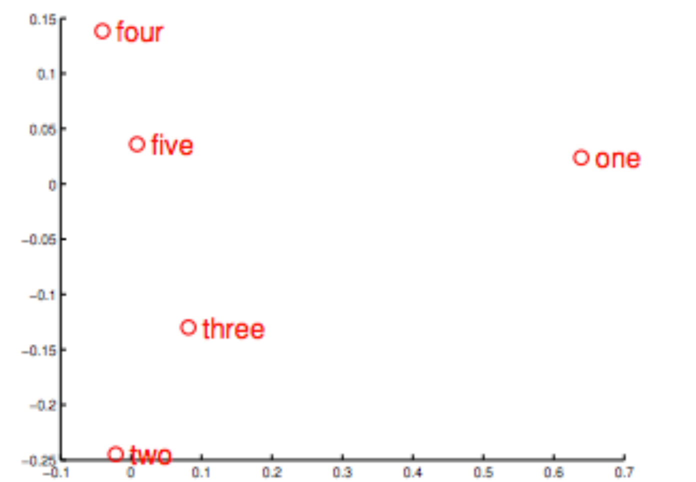
(also, Haghighi and Klein, 2008)

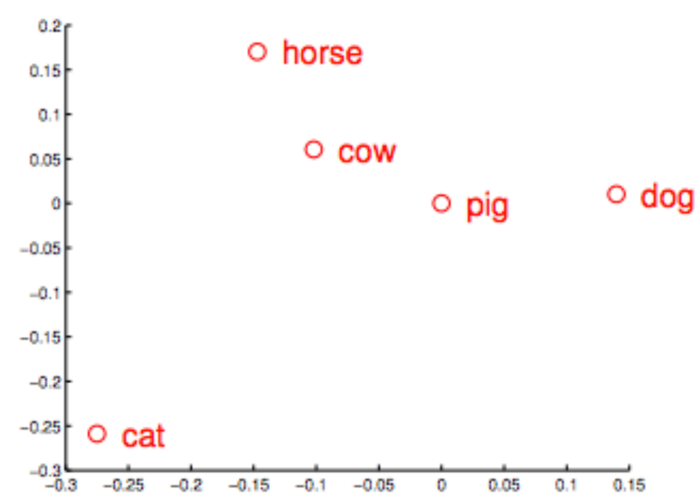
- “Exploiting Similarities among Languages for Machine Translation” - Mikolov, Le & Sutskever, 2013

- Observed a **similar structure in unsupervised embedding spaces of different languages**, after rotation

- Learned a rotation matrix to translate words from one embedding space to another with some success

- Weakly supervised - requires a small dictionary (5000 entries)

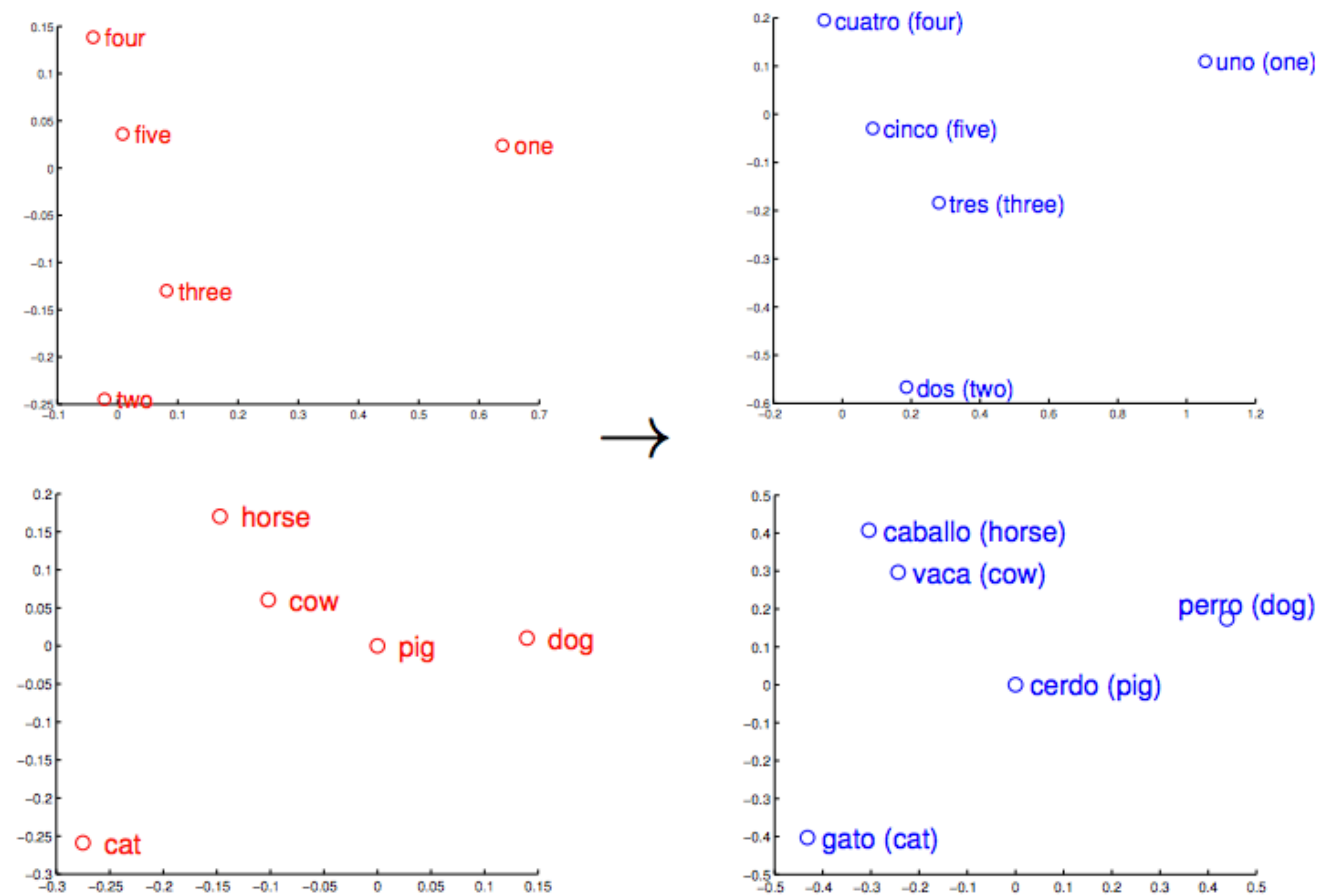




Motivation - Mikolov et al. 2013

(also, Haghighi and Klein, 2008)

- “Exploiting Similarities among Languages for Machine Translation” - Mikolov, Le & Sutskever, 2013
- Observed a **similar structure in unsupervised embedding spaces of different languages**, after rotation
- Learned a rotation matrix to translate words from one embedding space to another with some success
- Weakly supervised - requires a small dictionary (5000 entries)



Motivation - Mikolov et al. 2013

- Learned a rotation matrix to translate words from one embedding space to another with some success

$$\arg\min_{M} \sum_{x_i, y_i \in pairs} ||Mx_i - y_i||_2^2$$

Motivation - Mikolov et al. 2013

- Learned a rotation matrix to translate words from one embedding space to another with some success

$$\arg\min_{M} \sum_{x_i, y_i \in pairs} ||Mx_i - y_i||_2^2$$

$$= \arg\min_{M} ||ME^1 - E^2||_2^2$$

- Artetxe, Labake & Agirre, ACL 2017:
  - Use **numbers** as the initial pivot items.
  - Do it an an iterative procedure.

$$\arg\min_{M} \sum_{x_i, y_i \in pairs} ||Mx_i - y_i||_2^2$$

$$= \arg\min_{M} ||ME^1 - E^2||_2^2$$

- Artetxe, Labake & Agirre, ACL 2017:
  - Use **numbers** as the initial pivot items.
  - Do it an an iterative procedure.

$$\arg\min_{M} \sum_{x_i, y_i \in pairs} ||Mx_i - y_i||_2^2$$

$$= \arg\min_{M} ||ME^1 - E^2||_2^2$$

(also, can be solved exactly with SVD)

Beyond language-to-language

- Words from 1900 to words in 1990
- Words from young speakers to old speakers
- Words from left-wing to right-wing writers
- ....