Machine Learning

Representation learning

Dan Goldwasser

dgoldwas@purdue.edu

Some slides based on M. Gormley
NN Example slides

(Very) Modern Love

After my fiancé died, my mother told me to “get out there again.” She wanted me to go to a singles bar. I told her I’d rather go to the dentist.

“Just once,” she said. “Just to see what it’s like.”

One day, early last year, I found myself driving to a singles bar in winter snow. I sat in my car for 15 minutes, then drove away. The next day, I went back and sat in my car for another 15 minutes. I did this for a couple of weeks, until I finally mustered up the nerve to walk in.

https://www.nytimes.com/2020/11/24/science/artificial-intelligence-gpt3-writing-love.html

Language Models

- A language model over a given vocabulary V assigns probabilities to strings drawn from V*

Our goal is to assess whether

```
P(Private Customer… Be Toad)
>?<
P(Private Customer… Be Towed)
```

Language Models

- **A language model over a given vocabulary V assigns probabilities to strings drawn from V***

  Can we actually do it?

$$P_{ngram}(w_1 \ldots w_i) \; := \; P(w_1)P(w_2|w_1)\ldots P(\underbrace{w_i}_{nth\ word} \; | \underbrace{w_{i-n-1} \ldots w_{i-1}}_{prev.\ n-1\ words})$$

Unigram	$P(w_1)P(w_2)\ldots P(w_i)$
Bigram	$P(w_1)P(w_2\|w_1)\ldots P(w_i\|w_{i-1})$
Trigram	$P(w_1)P(w_2\|w_1)\ldots P(w_i\|w_{i-2}\ w_{i-1})$

Example: Trigram language model

- Consider the sentence:

Mr. Smith goes

$$p(\text{Mr. Smith goes STOP}) = p(\text{Mr.}|*,*))$$
$$p(\text{Smith}|*,\text{Mr.})\ \ p(\text{goes}|\text{Mr., Smith})\ \ p(\text{STOP}|\text{Smith, goes})$$

Model Estimation

- **How many parameters does the model need to estimate?**
  - Let's assume a trigram model, defined over vocabulary V
  - The number of parameters is $\mathbf{|V|^3}$
  - Let's assume: $|V| = 20K$ $< |V_{Shakespeare}|$
  - We'll have to estimate $\mathbf{8 \times 10^{12}}$ parameters
- **How many will we need to estimate for a unigram model?**
  - **Why not just do that?**

Language Models

Unigram	$P(w_1)P(w_2)\ldots P(w_i)$
Bigram	$P(w_1)P(w_2 \mid w_1)\ldots P(w_i \mid w_{i\text{-}1})$
Trigram	$P(w_1)P(w_2 \mid w_1)\ldots P(w_i \mid w_{i\text{-}2}\ w_{i\text{-}1})$

Our goal is to assess whether

P(Private Customer… Be Toad)

>?<

P(Private Customer… Be Towed)

What would be the answer if we use –

(1) a *Unigram* model? **(2)** a *Bigram* model?

Language Models

Unigram	• To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have • Every enter now severally so, let • Hill he late speaks; or! a more to leg less first you enter • Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like
Bigram	• What means, sir. I confess she? then all sorts, he is trim, captain. •Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. •What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman? •Enter Menenius, if it so many good direction found'st thou art a strong upon command of fear not a liberal largess given away, Falstaff! Exeunt
Trigram	• Sweet prince, Falstaff shall die. Harry of Monmouth's grave. • This shall forbid it should be branded, if renown made it empty. • Indeed the duke; and had a very good friend. • Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.
Quadrigram	• King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in; • Will you not tell me who I am? • It cannot be but so. • Indeed the short and the long. Marry, 'tis a noble Lepidus.

So how did we get here?

After my fiancé died, my mother told me to "get out there again." She wanted me to go to a singles bar. I told her I'd rather go to the dentist.

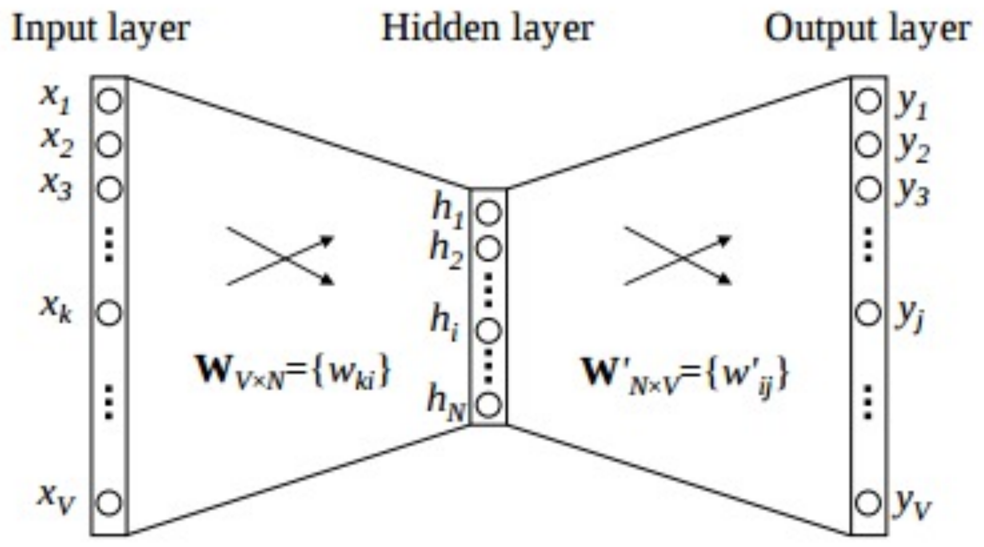
"Just once," she said. "Just to see what it's like."

One day, early last year, I found myself driving to a singles bar in winter snow. I sat in my car for 15 minutes, then drove away. The next day, I went back and sat in my car for another 15 minutes. I did this for a couple of weeks, until I finally mustered up the nerve to walk in.

https://www.nytimes.com/2020/11/24/science/artificial-intelligence-gpt3-writing-love.html

Word Embedding vs. AE

- Is this the same as an Auto-Encoder?



word2vec model architecture

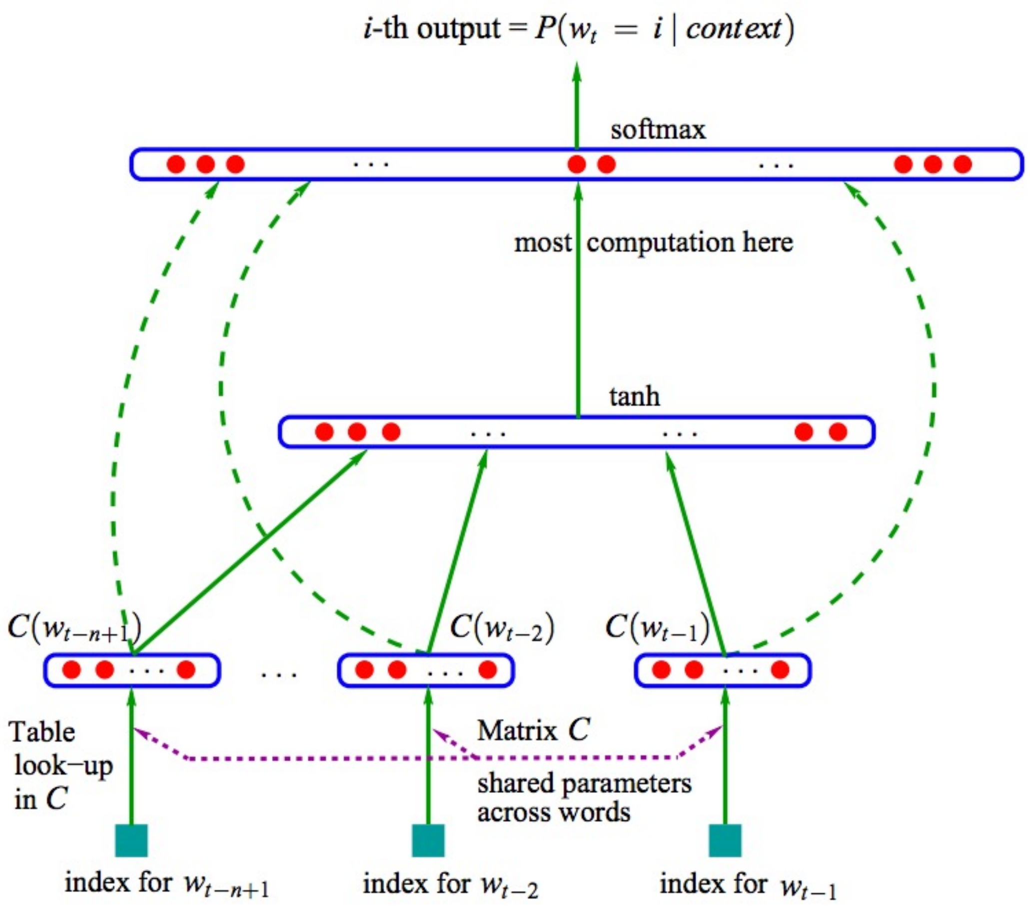
Reminder: *Language Models*

- A Language model defines a probability distribution over a sequence of words:

$$P(w_1, ..., w_n)$$

- Simple, yet very useful idea!
  - Estimate using a large collection of text (no supervision!)
  - **P("I like NLP") > P("me like NLP")**
- Key assumption: **Markov model**

Neural Language Model – Take 1

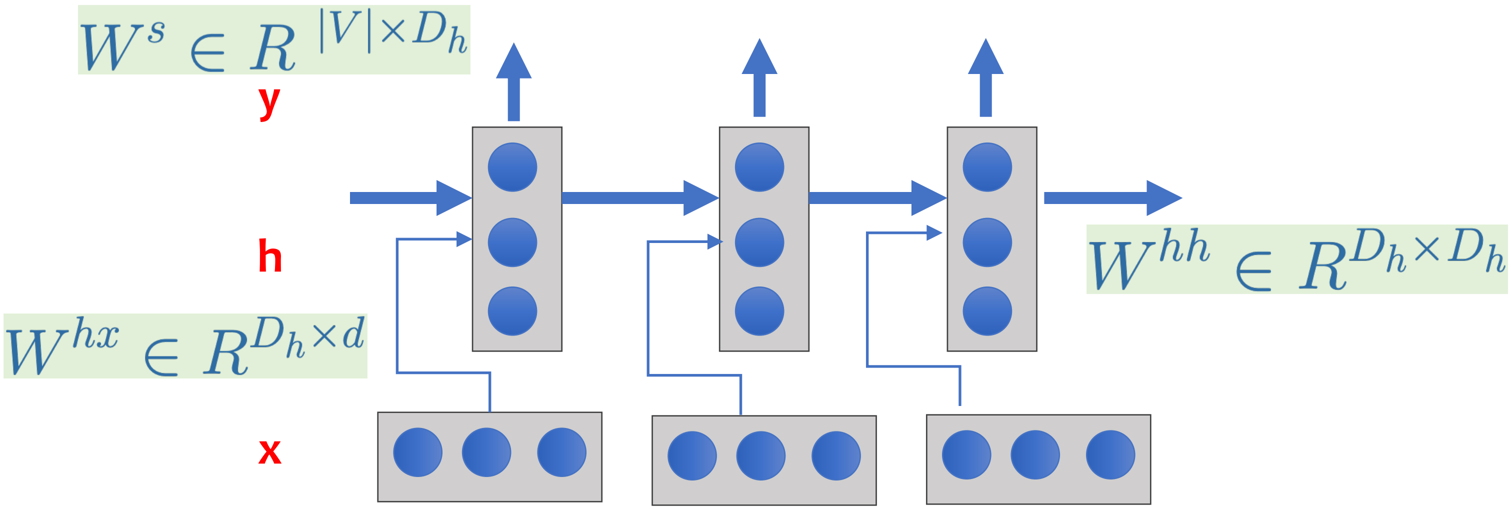


A Neural Probabilistic Language Model. Bengio et-al 2003

Recurrent Neural Networks

- A NN version of a language model.
  - More broadly: deal with **data over time.**
- Unlike N-gram models, an RNN conditions the current word on all previous words.
- **Efficient**, both in time and space

Recurrent Neural Networks



Input is a word (vectors) sequence: $x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n$

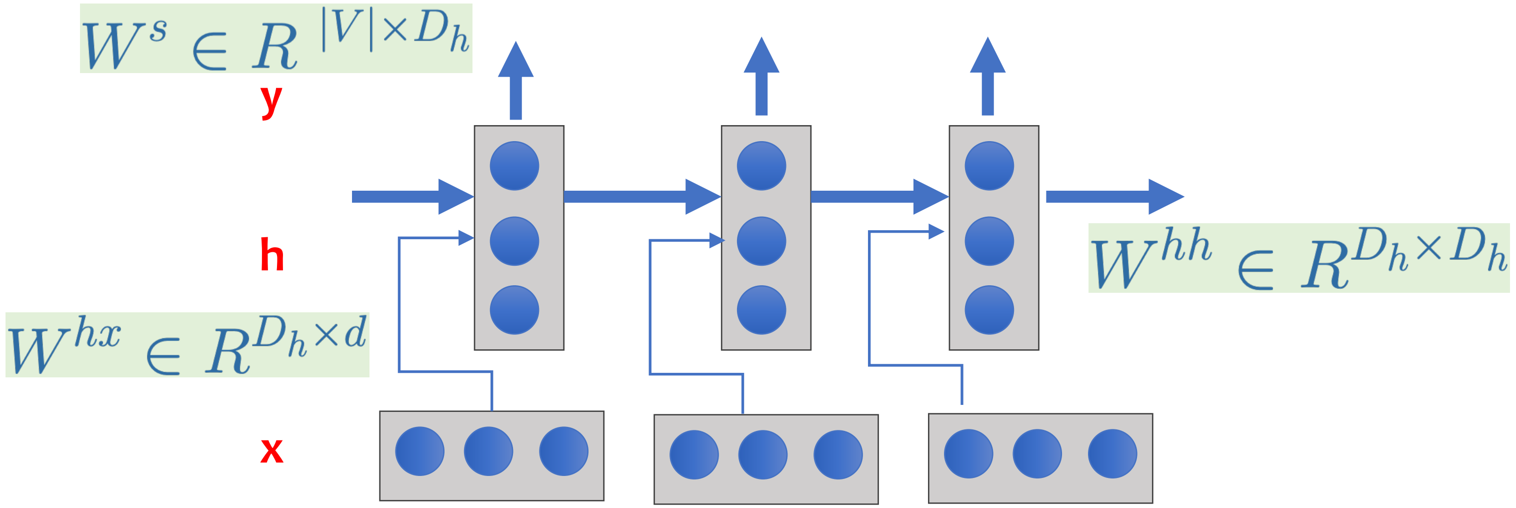
At any given time step *i* :

$$h_i = \sigma\left(W^{hh} h_{i-1} + W^{hx} x_i\right)$$

$$\hat{y} = \mathrm{softmax}(W^s\ h_i)$$

$$P(x_{i+1} = v_j | x_i, \ldots, x_1) = \hat{y}_{i,j}$$

RNN: Forward Propagation

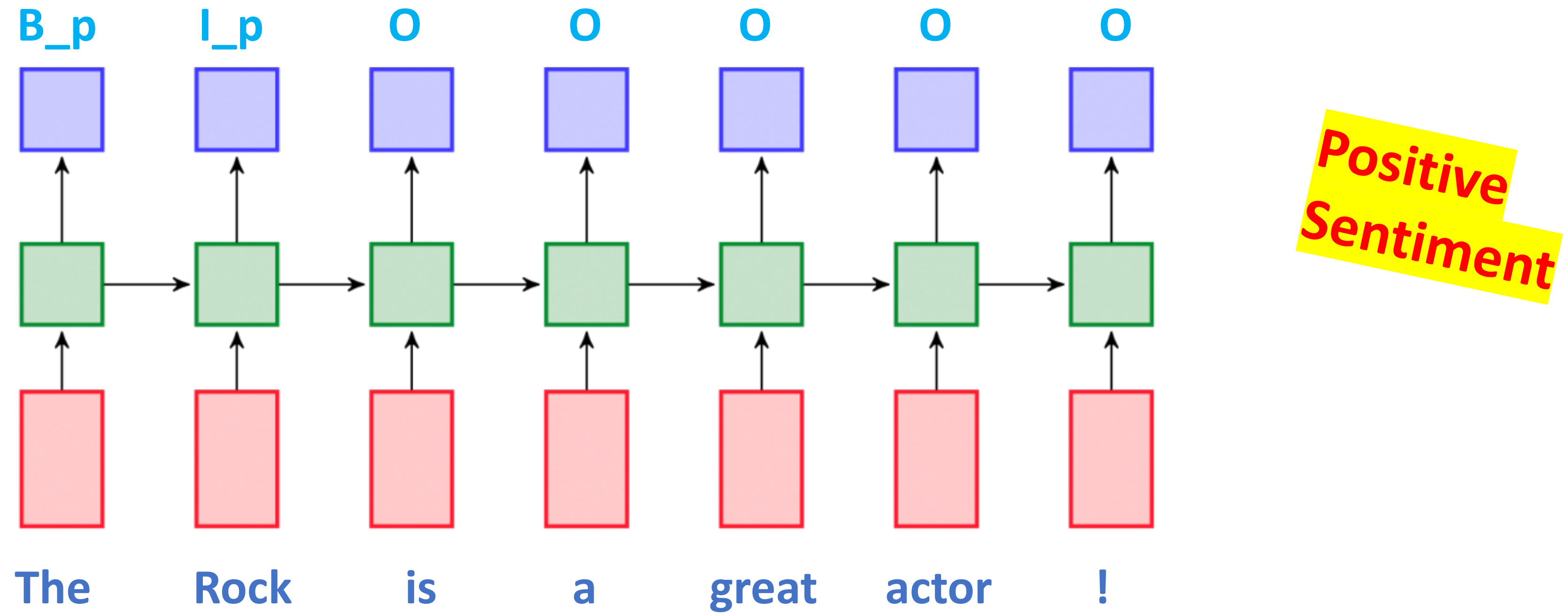


The cat sat on the mat. Where has the cat sat?

The cat sat on the hat. Where has the cat sat?

Beyond Language Models

- Recurrent architectures are extremely flexible.
  - They can be used as **text encoders**,
  - or as **sequence taggers**.



RNN Extensions

- **Key issue**: *long range dependencies between inputs.*
  - "how can we know which word is important to keep around, when predicting the i+1 word?"
- **Solution idea**: *complex hidden units that implement a "memory"*
  - Maintain "old memories" representing relevant long range dependencies
  - Error updates can be back-propagated at different strengths.

Gated Recurrent Units (GRU)

- Until now, we assumed a simple hidden layer:
  - representing the previous steps and input word

$$h_i = \sigma\left(W^{hh} h_{i-1} + W^{hx} x_i\right)$$

- In GRU's the picture is more complex, it adds **gates,** that control how the hidden state is computed
- *Essentially, more layers that can be learned from data*
  - **Update Gate**
  - **Reset Gate**

Gated Recurrent Units (GRU)

$$h_i = \sigma\left(W^{hh} h_{i-1} + W^{hx} x_i\right)$$

Original RNN

GRU:

Update Gate: $$z_i = \sigma\left(W^z x_i + U^z h_{i-1}\right)$$

Reset Gate: $$r_i = \sigma\left(W^r x_i + U^r h_{i-1}\right)$$

New memory $$\tilde{h}_i = \tanh\left(W x_i + r_i \circ U h_{i-1}\right)$$

Final memory (aka *hidden Layer*) $$h_i = z_t \circ h_{i-1} + (1 - z_i) \circ \tilde{h}_i$$

Why it works

- Learn a set of parameters for each one of the gates
  - **Recall**: *gates output a probability*
  - If **reset** gate is ~0: **ignore previous hidden state**
    - “forget” irrelevant information
    - **Short term dependencies**
  - If **update** gate ~1: **copy past information**
    - *“remember” past state*
    - **long term dependencies**

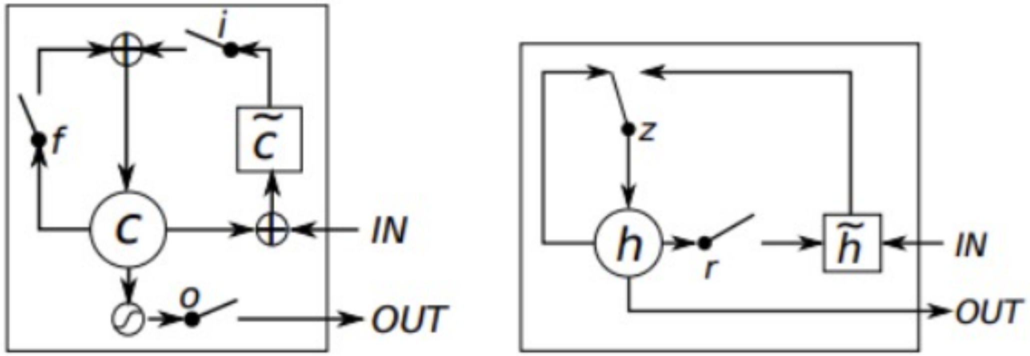
$$
\begin{aligned}
z_i &= \sigma\left(W^z x_i + U^z h_{i-1}\right) \\
r_i &= \sigma\left(W^r x_i + U^r h_{i-1}\right) \\
\tilde{h}_i &= \tanh\left(W x_i + r_i \circ U h_{i-1}\right) \\
h_i &= z_t \circ h_{i-1} + (1 - z_i) \circ \tilde{h}_i
\end{aligned}
$$

Long-Short-Term-Memories (LSTM)

- Similar (and older!) idea, though more complex
- Input gate $$i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1}\right)$$
- Forget gate $$f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1}\right)$$
- Output $$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1}\right)$$
- New memory $$\tilde{c}_t = \tanh\left(W^{(c)}x_t + U^{(c)}h_{t-1}\right)$$
- Final Memory $$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$
- Final hidden state $$h_t = o_t \circ \tanh(c_t)$$

Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Chung et-al 2014

RNN vs. LSTM vs. GRU



(a) Long Short-Term Memory (b) Gated Recurrent Unit

			tanh	GRU	LSTM
Music Datasets	Nottingham	train	3.22	2.79	3.08
		test	**3.13**	3.23	3.20
	JSB Chorales	train	8.82	6.94	8.15
		test	9.10	**8.54**	8.67
	MuseData	train	5.64	5.06	5.18
		test	6.23	**5.99**	6.23
	Piano-midi	train	5.64	4.93	6.49
		test	9.03	**8.82**	9.03
Ubisoft Datasets	Ubisoft dataset A	train	6.29	2.31	1.44
		test	6.44	3.59	**2.70**
	Ubisoft dataset B	train	7.61	0.38	0.80
		test	7.62	**0.88**	1.26

Table 2: The average negative log-probabilities of the training and test sets.

(Very) Modern Love

After my fiancé died, my mother told me to "get out there again." She wanted me to go to a singles bar. I told her I'd rather go to the dentist.

"Just once," she said. "Just to see what it's like."

One day, early last year, I found myself driving to a singles bar in winter snow. I sat in my car for 15 minutes, then drove away. The next day, I went back and sat in my car for another 15 minutes. I did this for a couple of weeks, until I finally mustered up the nerve to walk in.

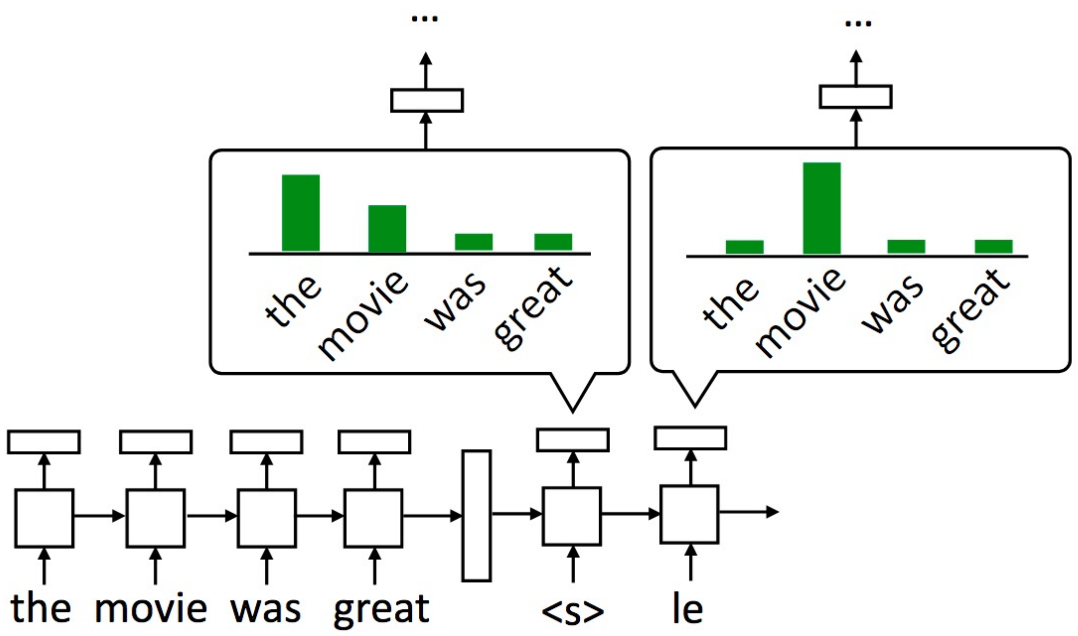
https://www.nytimes.com/2020/11/24/science/artificial-intelligence-gpt3-writing-love.html

Attention based "reading"

- Many of the challenges introduced by machine comprehension can be addressed using a general solution based on **attention**
- A general tool, currently used in **all** NLP tasks
  - Essentially, learn meaningful associations between inputs and outputs which can represent structural dependencies.

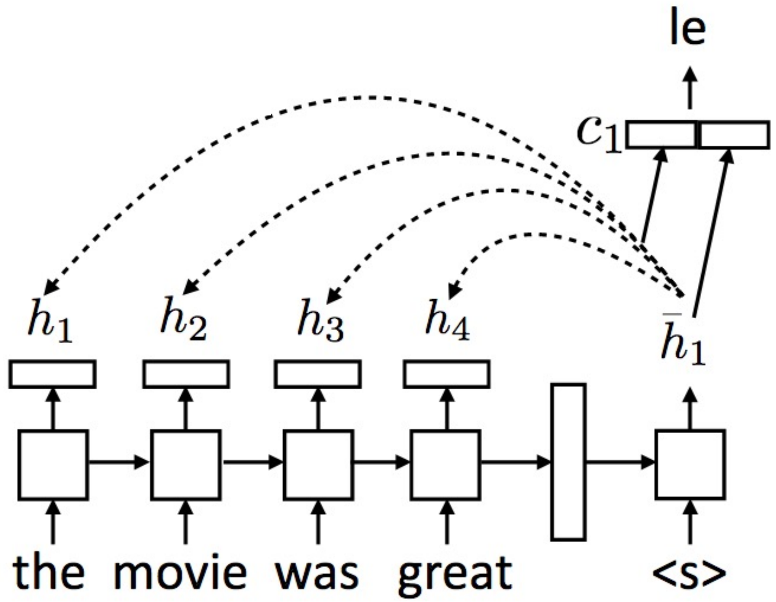
Attention

- **Attention**: at each decoder state computes a **distribution over the source inputs** based on the current decoder state



Attention

- For each decoder state compute the weighted sum of input states



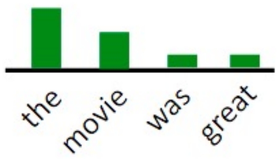
Decision at step i:

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

$$c_i = \sum_j \alpha_{ij} h_j$$

- Weighted sum of input hidden states (vector)

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

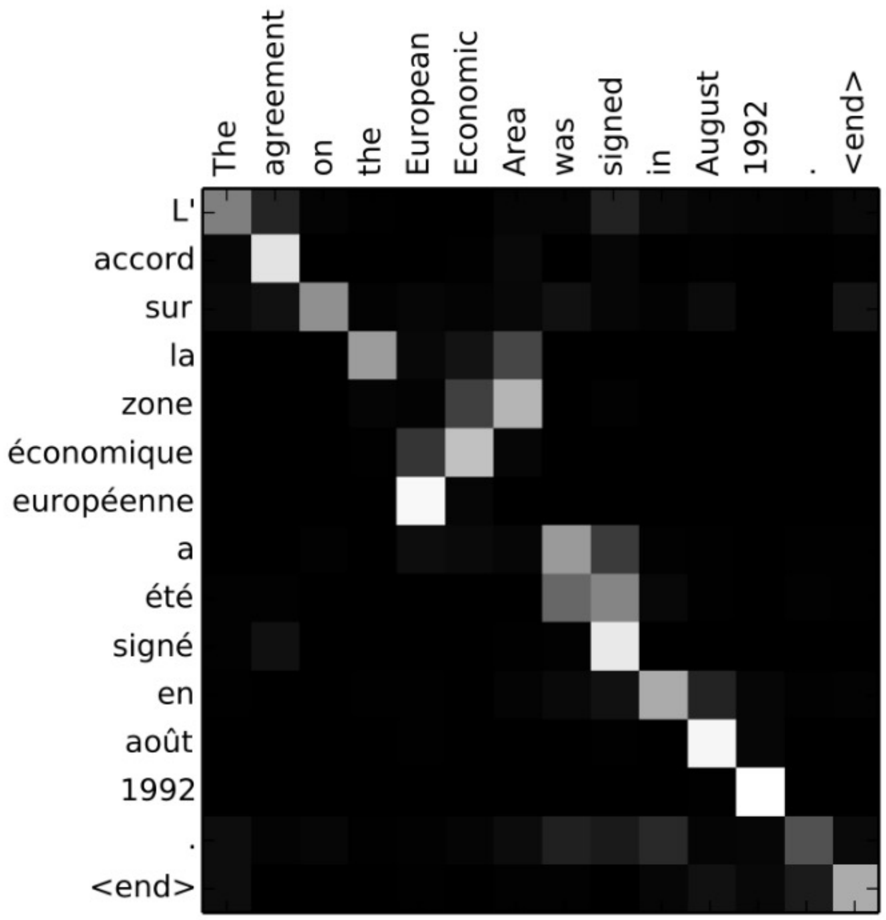


$$e_{ij} = f(\bar{h}_i, h_j)$$

- Unnormalized scalar weight

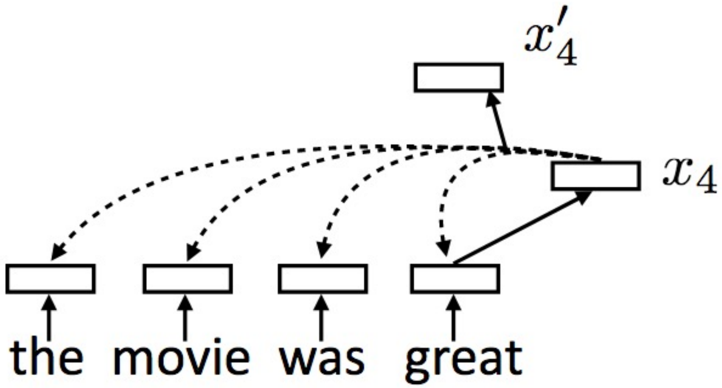
Attention

We can identify the source word context for output predictions



Self-attention

- **A new way to represent structure**
  - *Each word forms a query which computes attention over each word*



$$\alpha_{i,j} = \text{softmax}(x_i^\top x_j)$$ scalar

$$x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j$$ vector = sum of scalar * vector

The representation of each word is a function of its neighbors. Does that sound familiar?

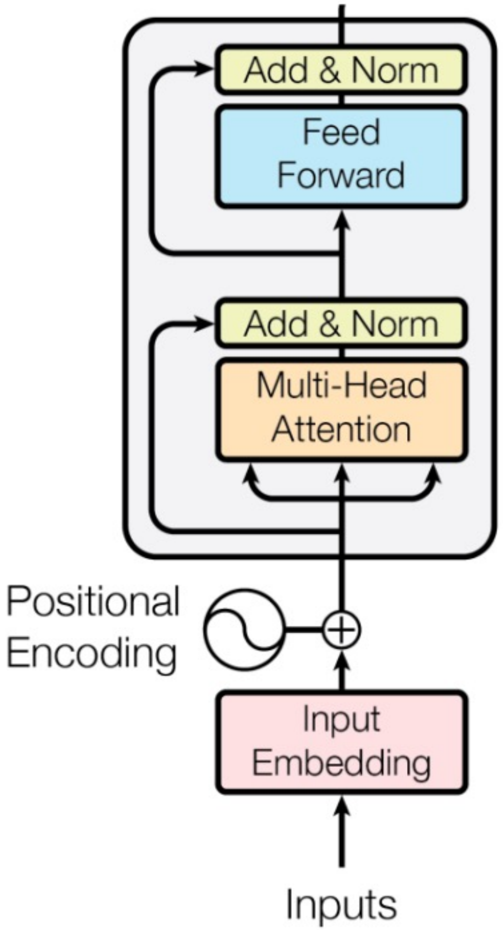
Vaswani et al. (2017)

Transformers

- The idea of self attention was extremely influential in NLP
  - **No fixed position representation as in LSTM instead structure is represented the attention assignments.**

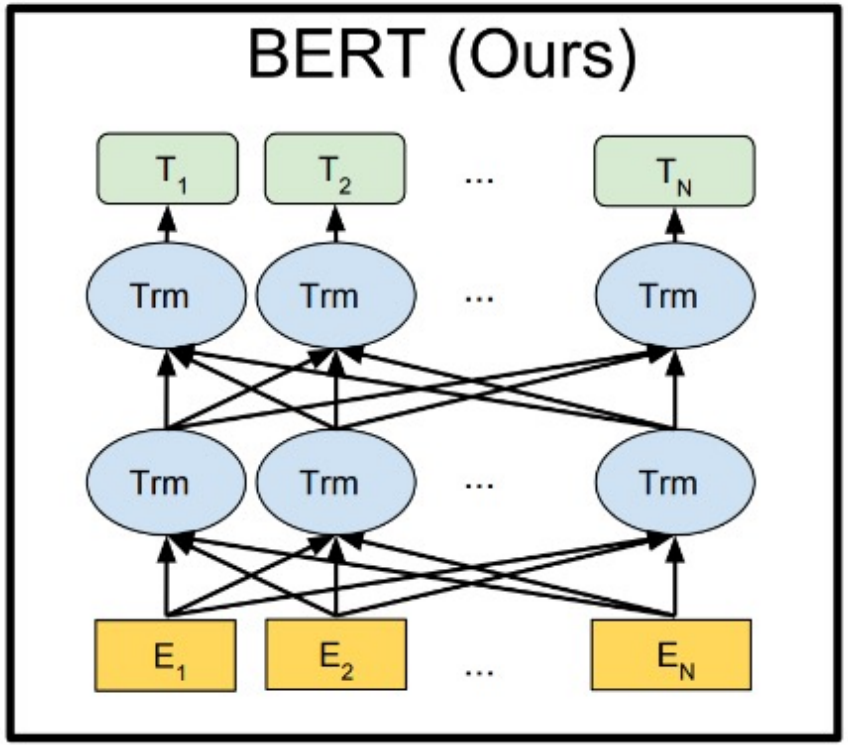
I like bananas but not carrots.
Vs.
I like carrots but not bananas

- In reality, position information is needed, but it is used differently compared to an LSTM, by encoding it as part of the input



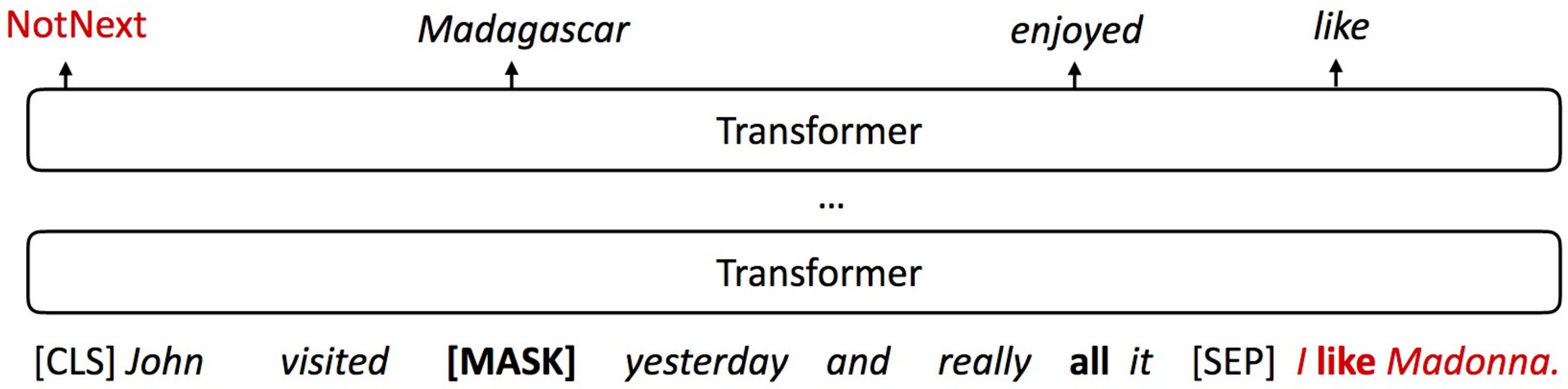
BERT

- **Transformer-based approach** *instead of an LSTM-based like ELMo*.
  - Transformer vs. LSTM
  - Masked language objective instead of usual LM
  - Fine-tuned at test time



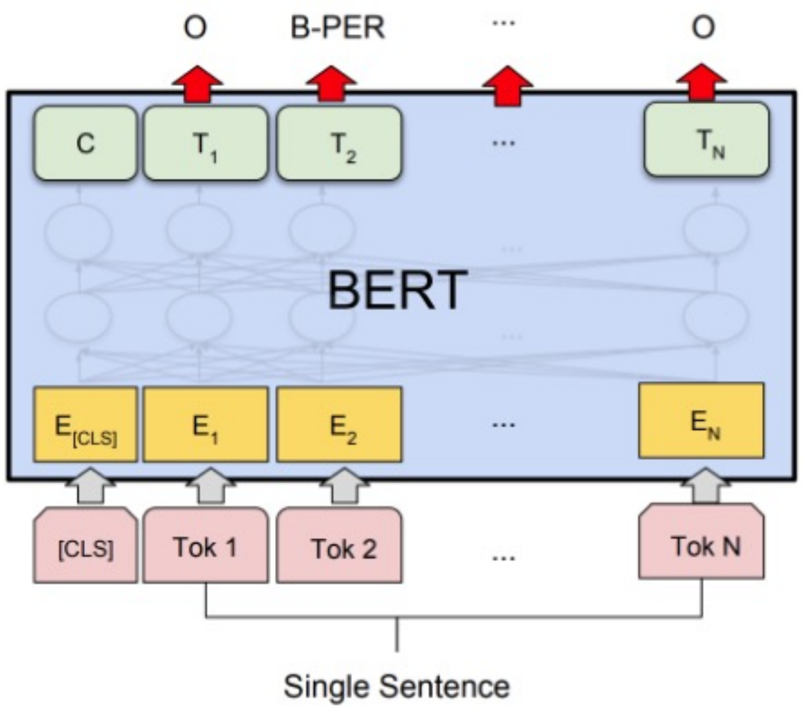
Next sentence

- **BERT objective**: masked LM + next sentence

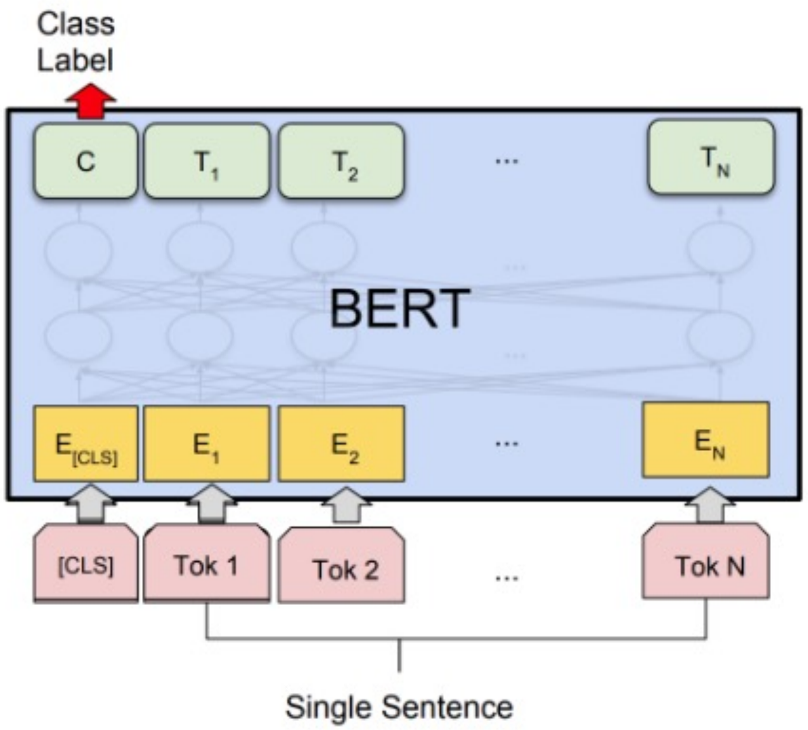


BERT in practice

Very flexible, can be used for NLI, classification, tagging, etc.

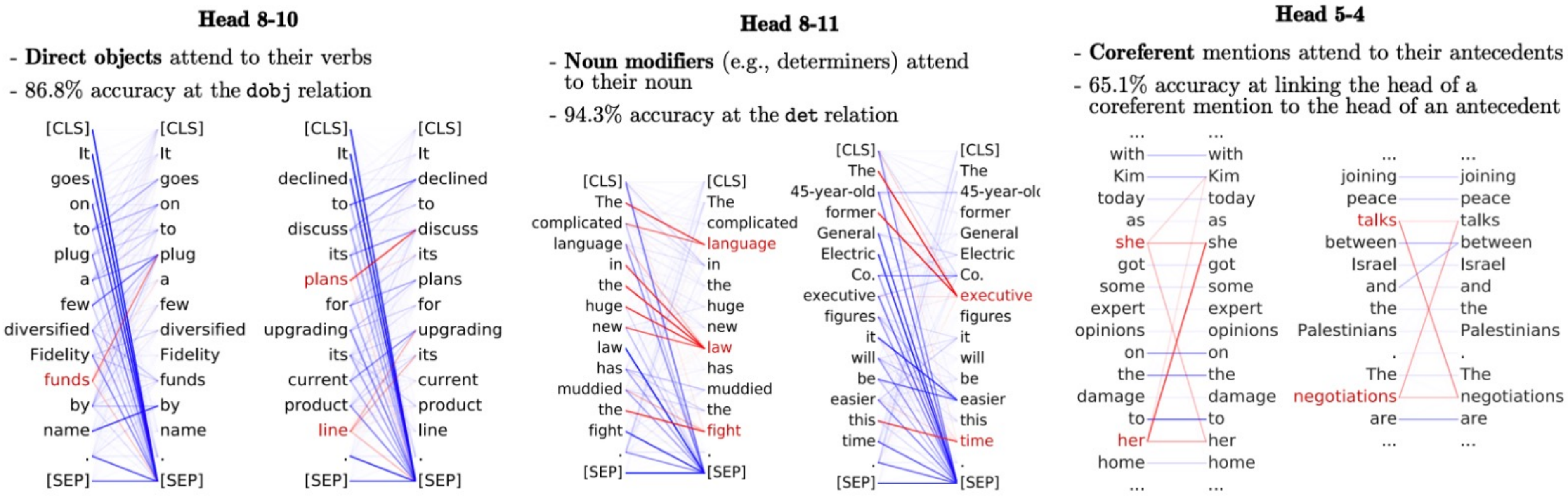


(d) Single Sentence Tagging Tasks: CoNLL-2003 NER



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

What does BERT learn?



- Still way worse than what supervised systems can do, but interesting that this is learned organically

Discussion

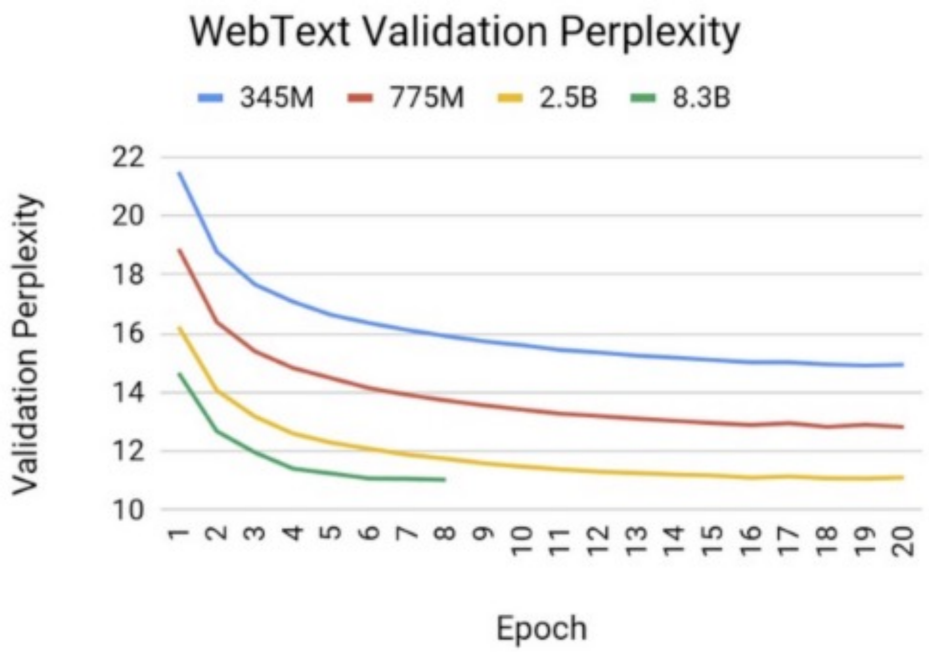
- **Current NLP trend**: *train a **very complex** neural language model using **massive** amounts of data*
- The learned representation should capture ”language understanding capability”
  - Word meaning
  - Linguistic structure
  - World knowledge
  - Bad stuff expressed through language.

Language Modeling

- *Impossible, **but**.. Bigger models do better and better*
  - Limit not yet reached

‣ NVIDIA: trained 8.3B parameter GPT model (5.6x the size of GPT-2)

‣ Arguable these models are still underfit: larger models still get better held-out perplexities



NVIDIA blog (Narasimhan, August 2019)

Open Questions

- Language modeling as a pre-training, and transformer-based architectures are now SoTA for all major NLP task.
  - GPT-3 uses 175 Billion parameters.

  That's a lot!

  Is this a sustainable direction? Currently all models developed in industry

  Building these models in academia would require a collective effort, might require changing how research is done.

Open Questions

- **What is really learned by a language model and how should we think about them?**
  - It’s clear that our original definition was too simplistic (capture word-patterns for tasks like context sensitive spelling correction)
  - Do we still need symbolic knowledge graphs? Can everything ”just” be learned from raw text using a language model?

Evaluating Learned Representations

Q: What is human life expectancy in the United States?
A: Human life expectancy in the United States is 78 years.

Q: Who was president of the United States in 1955?
A: Dwight D. Eisenhower was president of the United States in 1955.

Q: What party did he belong to?
A: He belonged to the Republican Party.

GPT 3 Generated Answers

Q: Which is heavier, a mouse or an elephant?
A: An elephant is heavier than a mouse.

Q: Which is heavier, a mouse or a toaster?
A: A toaster is heavier than a mouse.

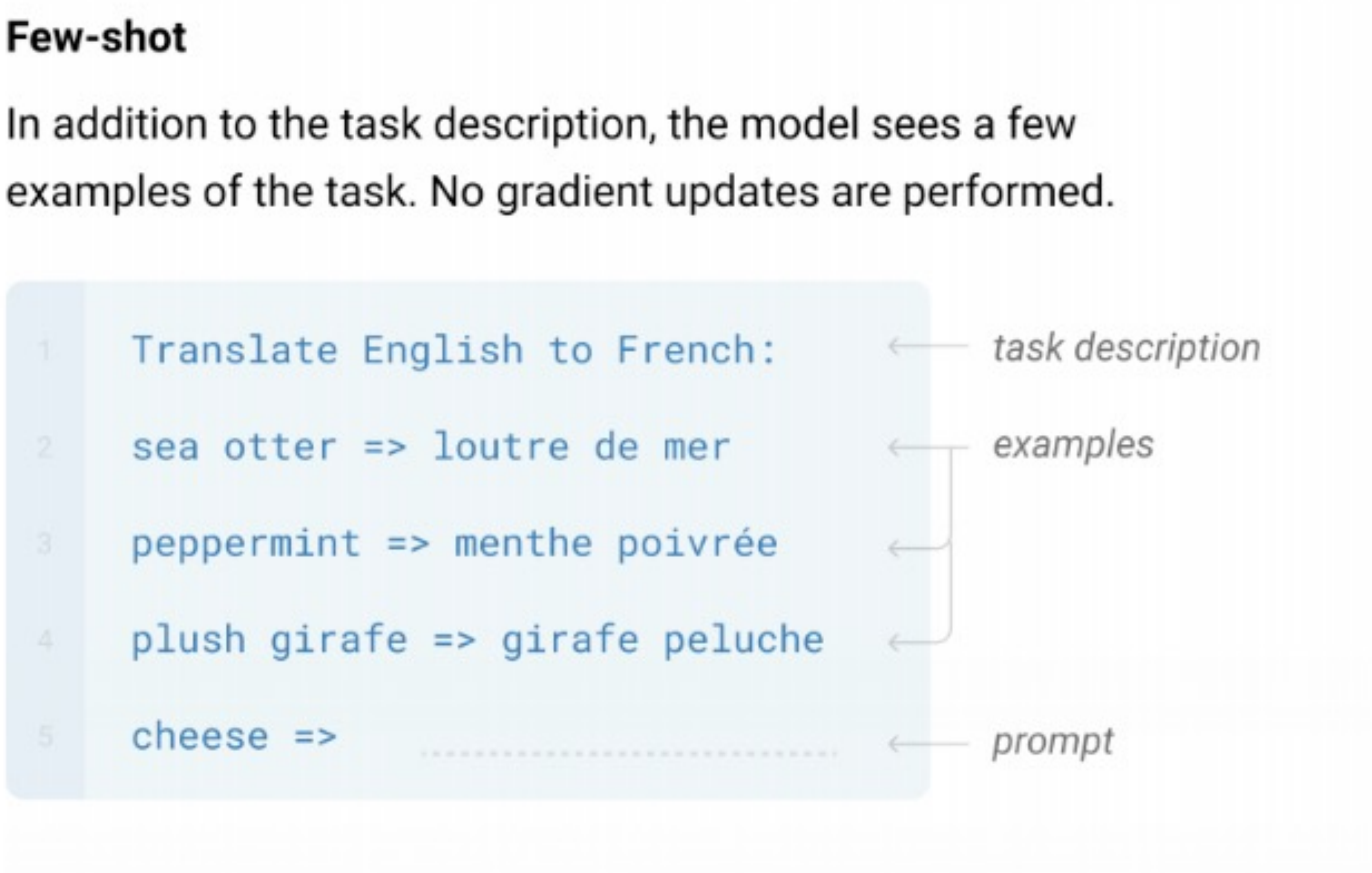
Q: Which is heavier, a toaster or a pencil?
A: A pencil is heavier than a toaster.

https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

Evaluating Machine Comprehension

- The models we discussed are very flexible.
  - Highly expressive models, trained over massive amounts of text.
  - **What do they actually learn?**
  - Memorize facts, capture common-sense judgements, world knowledge, etc.
- Many tasks can be formulated as LM
  - Question answering, discourse relation, etc.

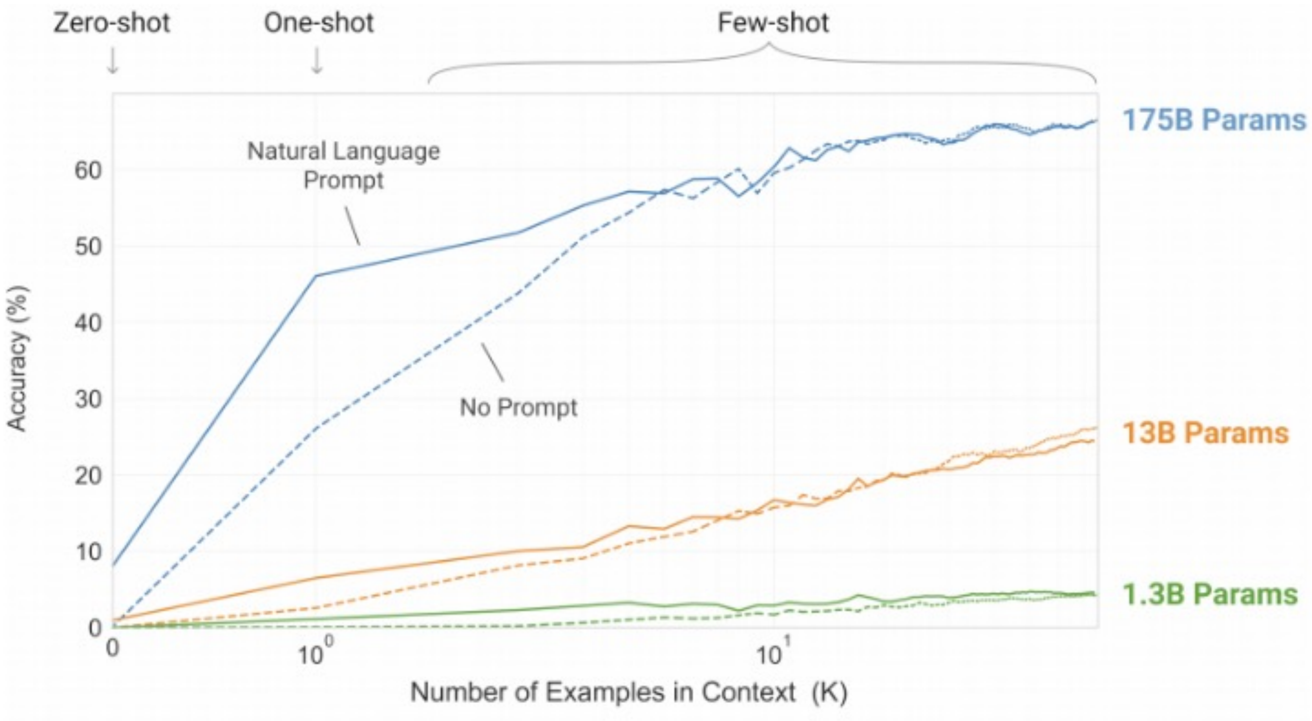
GPT3: new directions



Brown et al. (2020)

GPT3: new directions

- **Key observation:** few-shot learning only works with the very largest models!



Brown et al. (2020)