Applied Machine Learning!!!

W207 Section 9
Rasika Bhalerao
rasikabh@berkeley.edu

Aug 23: Welcome!
Nov 8 and 22: No classes

Schedule

Supervised learning methods

	Sync	Topic
2	Aug 30	Linear Regression / Gradient Descent
3	Sep 6	Feature Engineering Bonus: Naive Bayes
4	Sep 13	Logistic Regression
5	Sep 20	Multiclass classification / Eval Metrics Bonus: Reinforcement learning
6	Sep 27	Neural Networks
7	Oct 4	KNN, Decision Trees, Ensembles

Unsupervised learning methods

	Sync	Topic
8	Oct 11	KMeans and PCA
9	Oct 18	Text Embeddings Bonus: Language models
10	Oct 25	CNNs Bonus: GANs
11	Nov 1	EDA, Real data, Baselines, LDA
12	Nov 15	Fairness / Ethics
13	Nov 29	Fancy Neural Networks
14	Dec 6	Final Presentations

Behavior expectations

- Healthy disagreement is expected
- Be mindful of one another’s schedules
- Be a good listener
- Have fun in a professional manner
- Share related real-world experience
- Ask questions when something is confusing
- Keep it 100 but be respectful
- Be open-minded to new ideas in the real world and when coding
- On time for group meetings

How are final projects going?

Guidelines: https://docs.google.com/document/d/1R7mIHOtYXKU8vEQzw10uofb_iK3sgimw8iZLWSTzdgg/edit?usp=sharing

Sequences

- Non-sequential machine learning takes input from a single point in time, and outputs a prediction or classification:

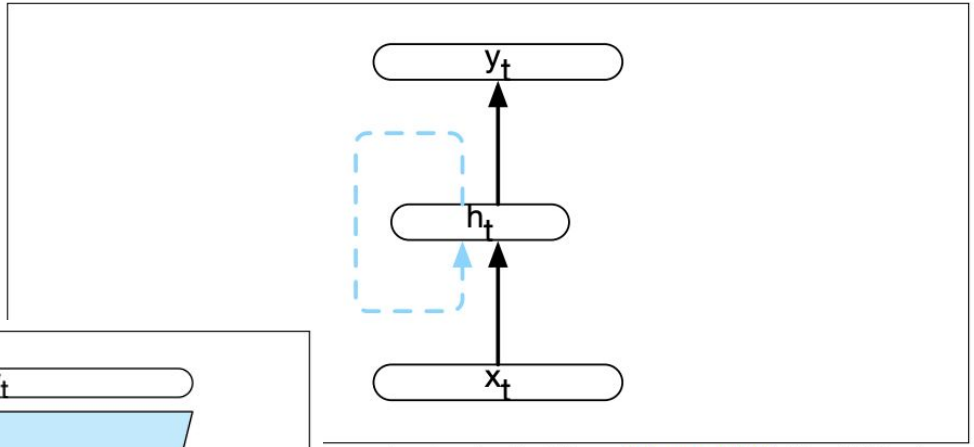
  $y = f(x(t))$

- Now we look at sequential inputs where the output y can depend on more than just the immediate input:

  $y = f(s(t)) = F(x(t), x(t-1), \ldots, x(1))$

Solution: Recurrent Neural Network

- The hidden state can loop back to itself to use again with the next term in the sequence
- No fixed length



neural network after Elman (Elman, 1990). The hidden layer
ı as part of its input. That is, the activation value of the hidden
ıput as well as the activation value of the hidden layer from the

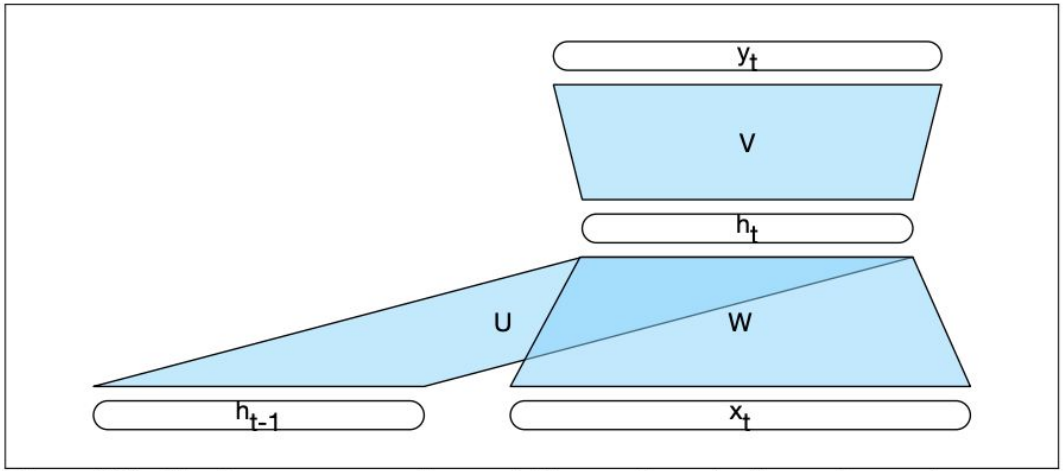


Figure 9.3 Simple recurrent neural network illustrated as a feedforward network.

Training a Neural Network

- **Forward pass**: process the sequence
- **Backward pass**: Backpropagation Through Time
- Note: since the input at each step affects multiple outputs (and therefore different parts of loss), we “assign blame” proportionally to weights when updating weights in gradient descent

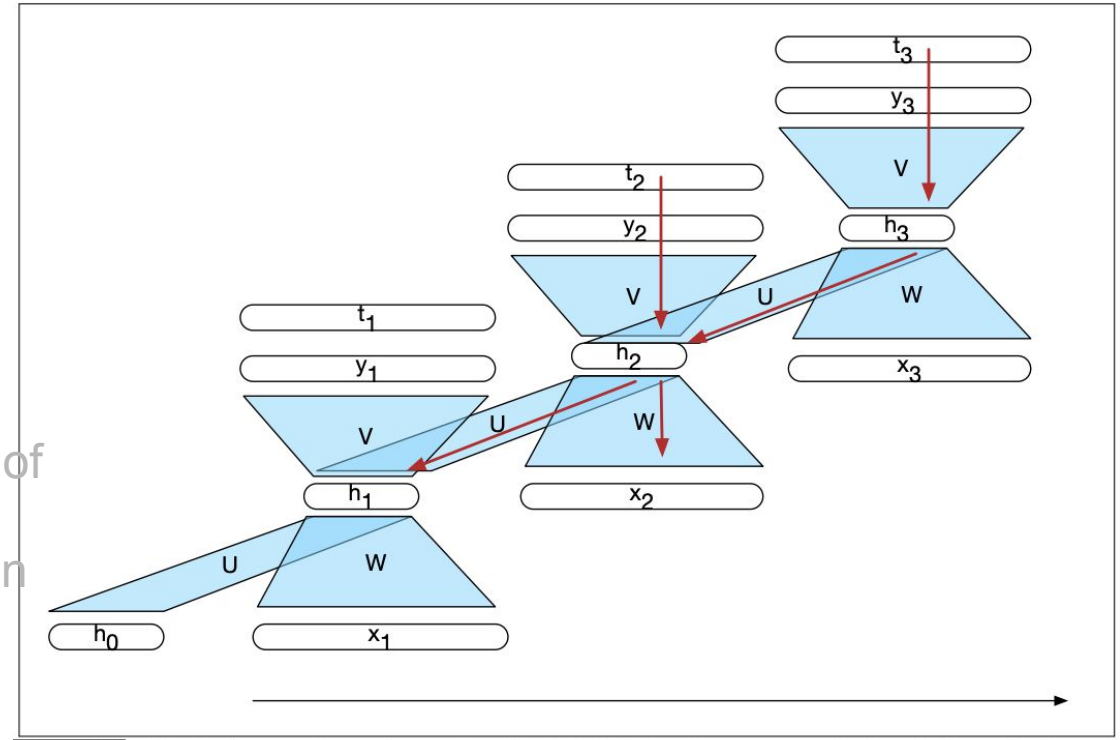


Figure 9.6 The backpropagation of errors in a simple RNN t_i vectors represent the targets for each element of the sequence from the training data. The red arrows illustrate the flow of backpropagated errors required to calculate the gradients for U, V and W at time 2. The two incoming arrows converging on h_2 signal that these errors need to be summed.

Generation with Recurrent Neural Language Models

- Called autoregressive generation
- Shannon's Method: do generation the same way we did with Ngram language models, where we start with <s> and randomly pick the next word using the probabilities until we pick </s> (or reach a fixed length limit)
- Machine translation, summarization, and question answering work by using a variation of this, but with the last hidden state of the "encoder" RNN instead of <s>

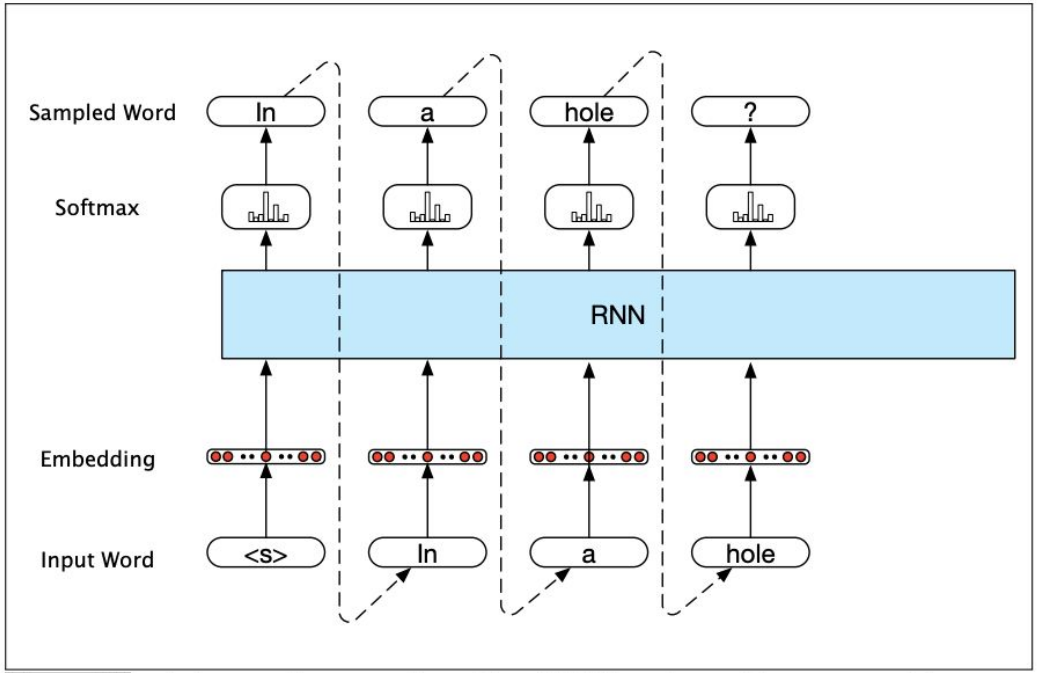


Figure 9.7 Autoregressive generation with an RNN-based neural language model.

Sequence Labeling with RNNs

- Task: label each input word (e.g. POS tags)
- Inputs: word embeddings
- Outputs: tag / label for each word
- Common way to draw RNN →

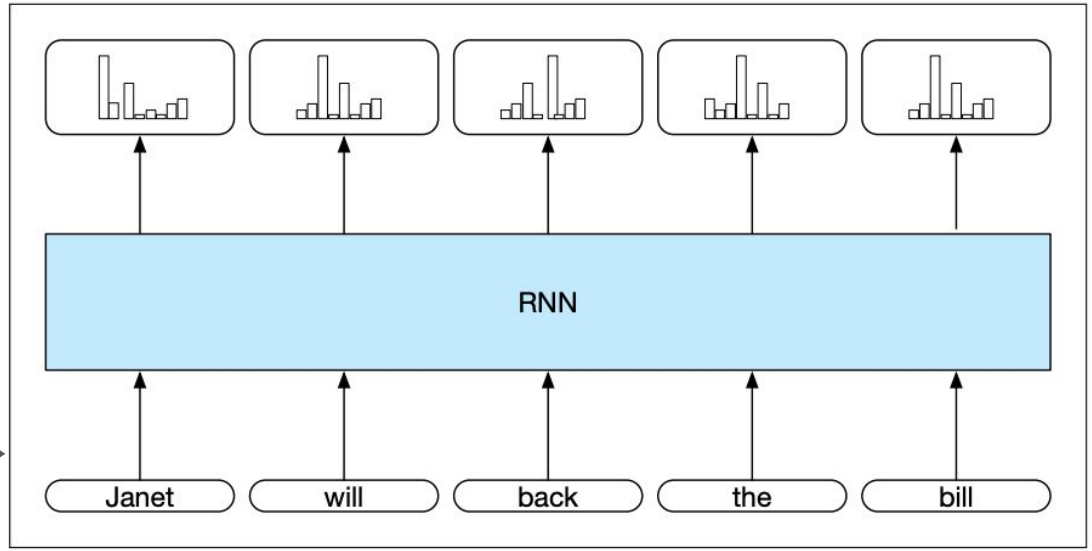
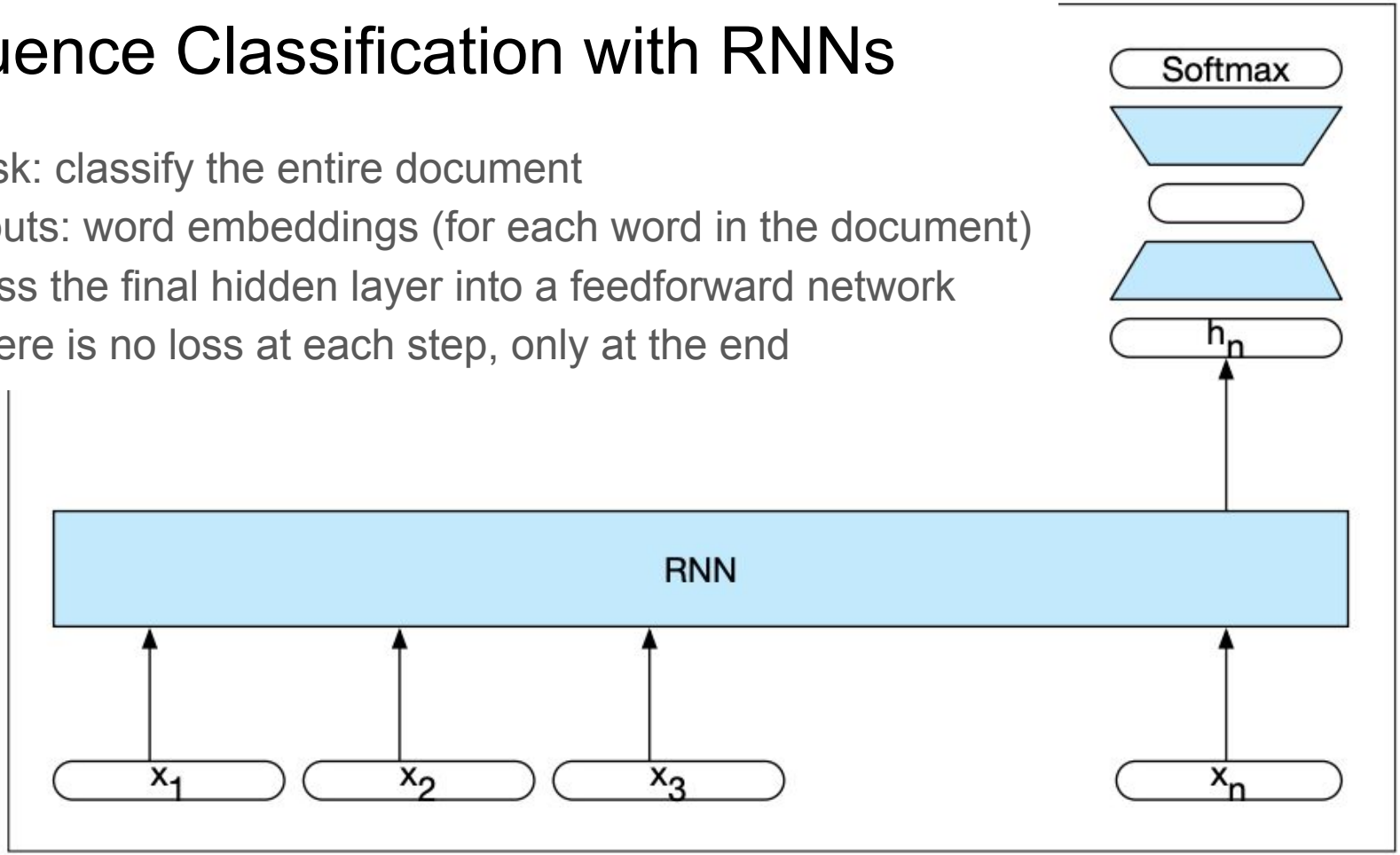


Figure 9.8 Part-of-speech tagging as sequence labeling with a simple RNN. Pre-trained word embeddings serve as inputs and a softmax layer provides a probability distribution over the part-of-speech tags as output at each time step.

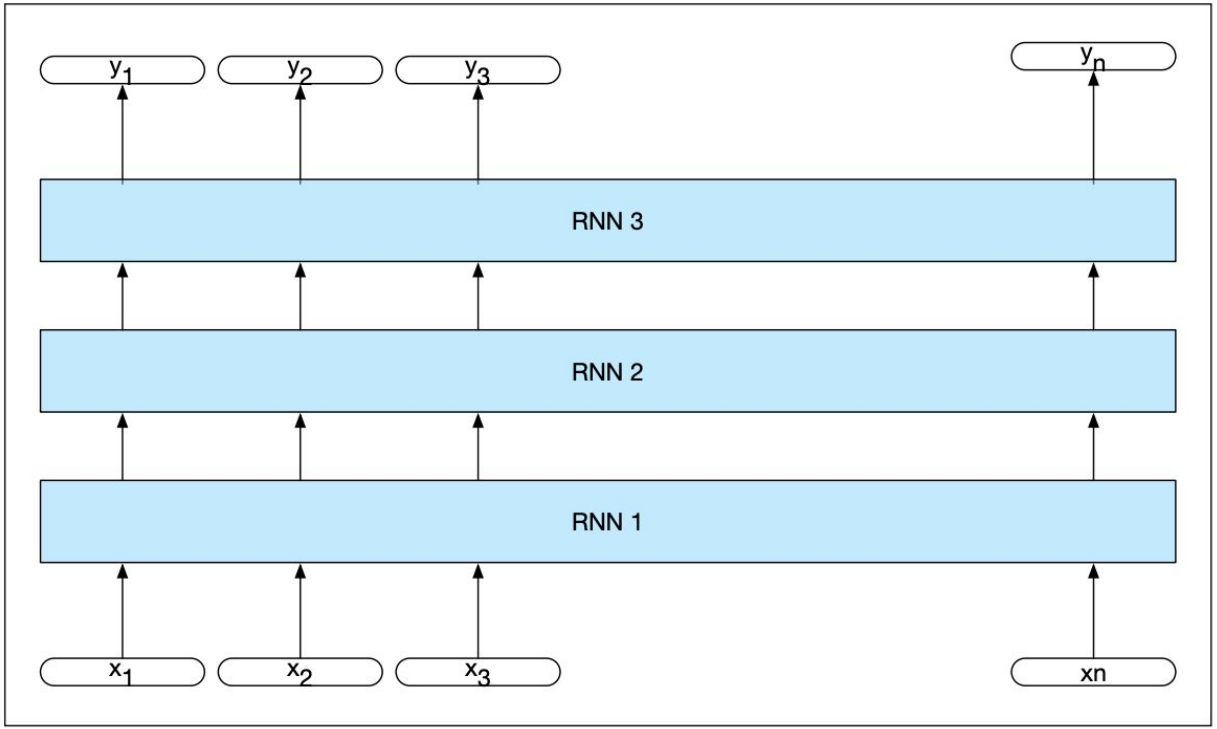
Sequence Classification with RNNs

- Task: classify the entire document
- Inputs: word embeddings (for each word in the document)
- Pass the final hidden layer into a feedforward network
- There is no loss at each step, only at the end



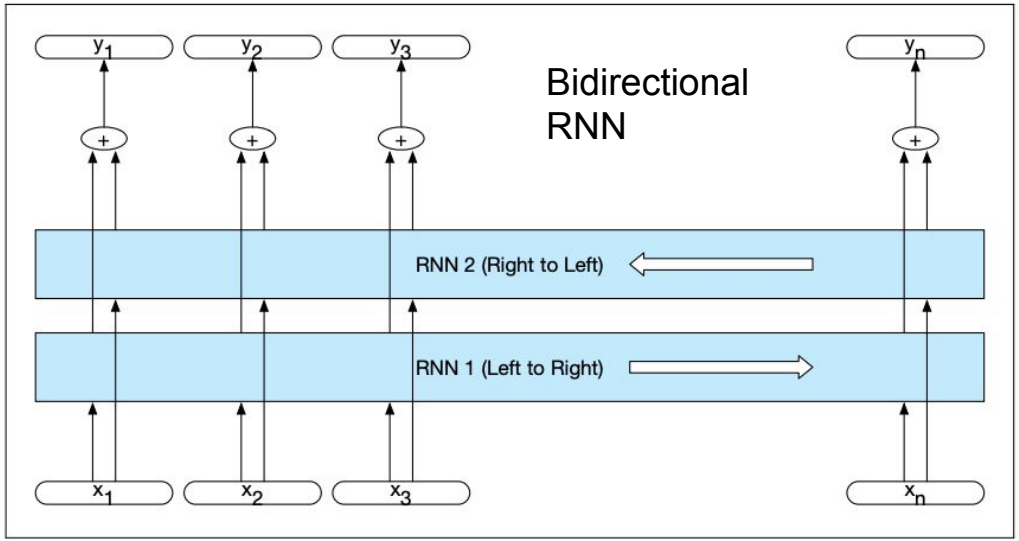
Stacked RNNs

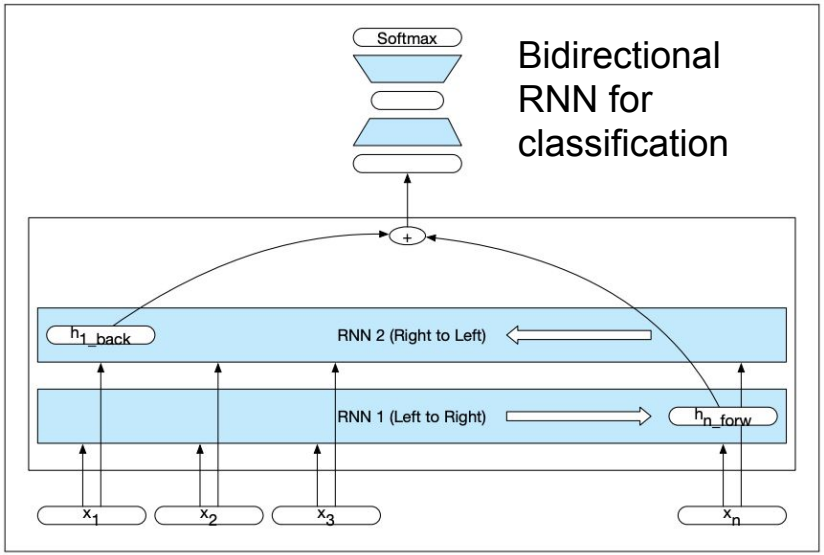
- Take the sequence of hidden layers as input to another RNN
- This has been shown to help detect layers of abstraction (the same way our eyes and conv. neural networks use pixels to detect edges and then use edges to detect shapes)
- This is slow



Bidirectional RNNs

- One RNN goes forward as usual
- Another RNN starts from the last word and goes backwards
- Hidden layer at each step = combination of the hidden state from the forward and backward RNNs
    - Combine using concatenation, addition, multiplication, or averaging





Long Short-Term Memory (LSTM)

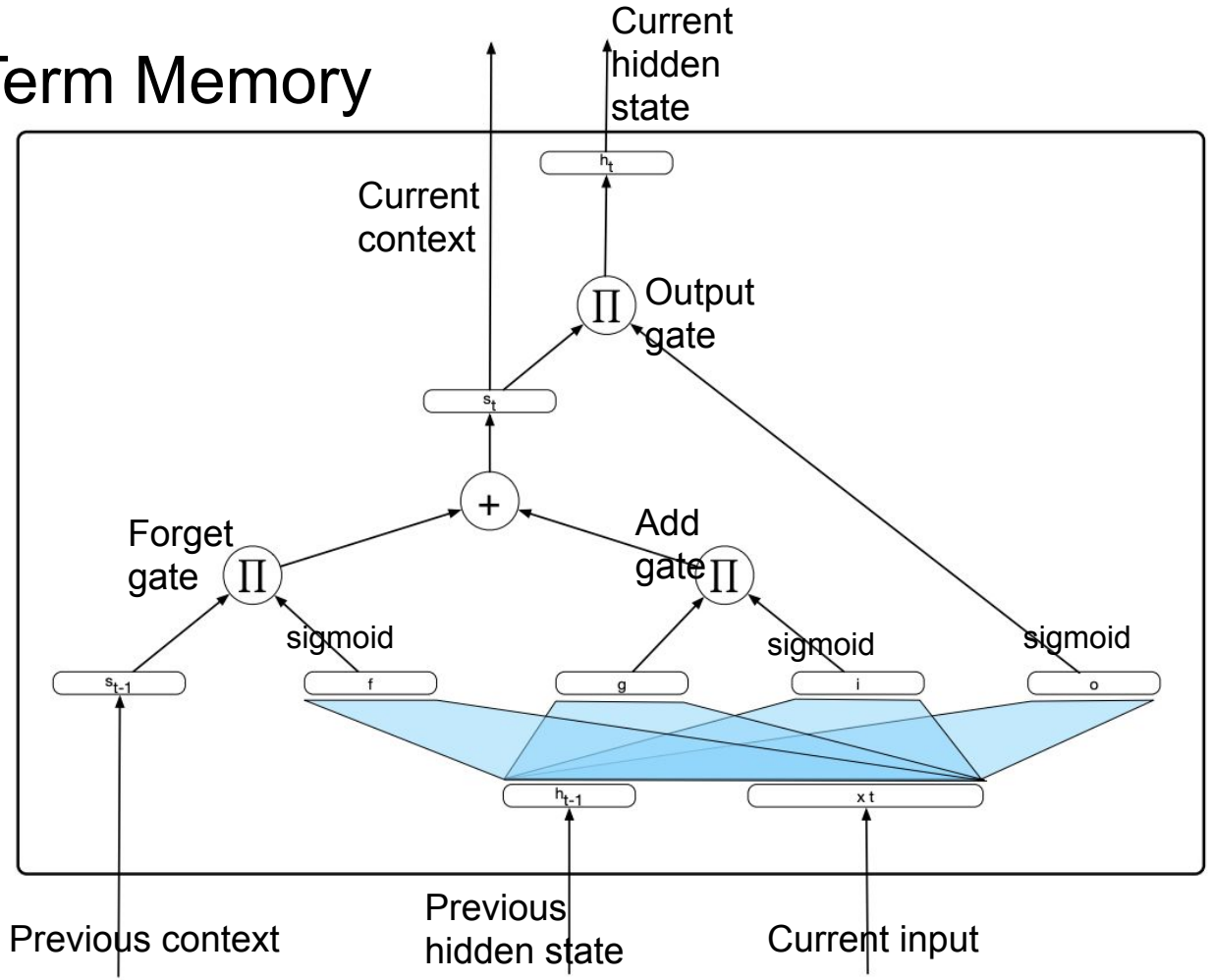
- Vanishing Gradients problem: with a regular RNN, due to the multiplications in the chain rule, words at the beginning of a sequence have less of an impact on the output than words at the end of the sequence
- This is why we often use an **LSTM**
- We address this by adding a context layer and three “gates”
  - Each gate has a feedforward layer, a sigmoid activation, and then pointwise multiplication with the layer being gated
  - The sigmoid almost turns the output of the feedforward layer into a binary mask (sigmoid pushes numbers close to 0 and 1)
  - This helps tell the network which elements of the layer being gated are important to remember
  - The gates are the forget gate, add gate, and output gate

Long Short-Term Memory

g is the part that came from RNN (with a tanh activation function)

f, i, and o are sigmoided and used as the masks for the forget, add, and output gates, respectively

Output gate decides what is important for current output as opposed to later



Subwords

Downsides to representing entire words as embeddings:

- Some lexicons have too many words
- Unknown words (rare words and misspellings)
- Morphological information is useful

Approaches to addressing this:

- Character ngrams
- Subwords based on phonetic analysis
- Morphological analysis to get linguistically motivated word pieces (usually overkill)

RNNs with Subwords

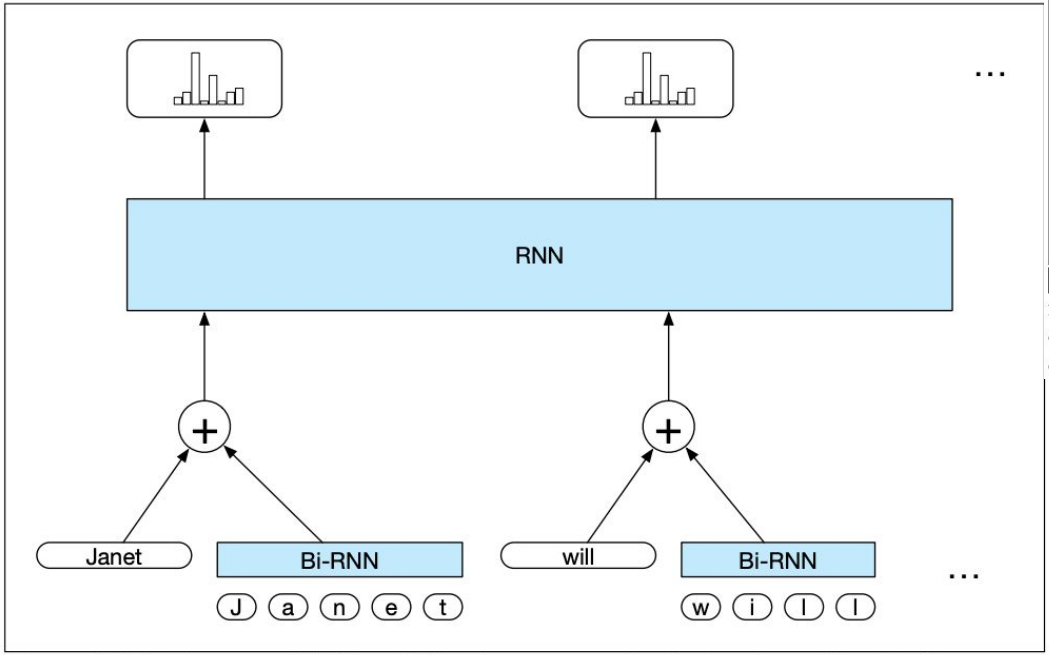


Figure 9.15 Sequence labeling RNN that accepts distributional word embeddings augmented with character-level word embeddings.

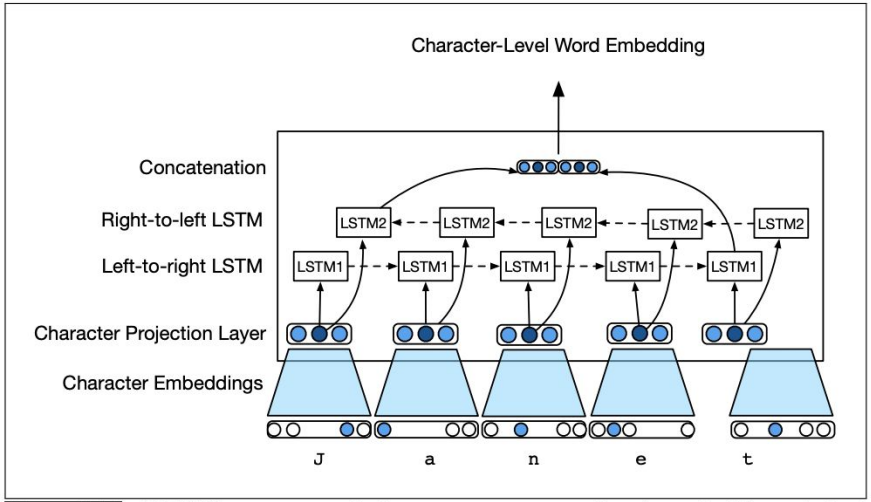


Figure 9.16 Bi-RNN accepts word character sequences and emits embeddings derived from a forward and backward pass over the sequence. The network itself is trained in the context of a larger end-application where the loss is propagated all the way through to the character vector embeddings.

The lower Bi-RNN processes sequences of characters, and the upper RNN processes sequences of words.

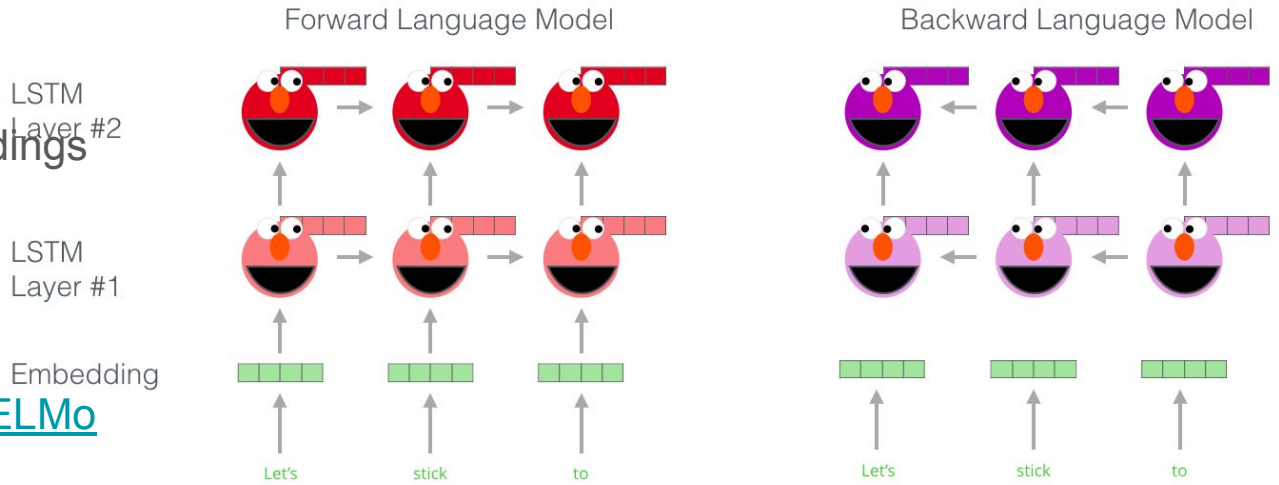
This way you can still get output per word.

The backprop goes from the output task all the way back to the character level.

RNNs can be any variation (LSTM, Bi-RNN, …)

ELMo

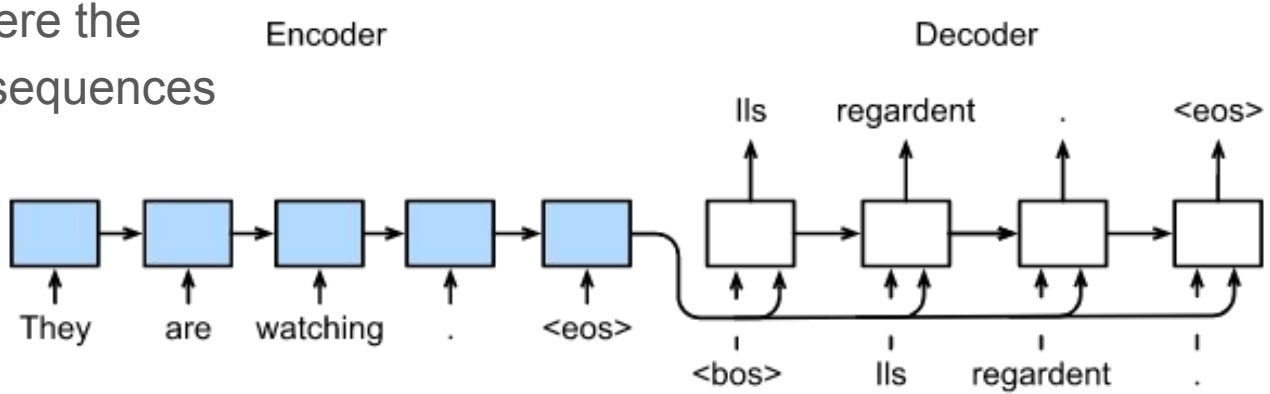
- ELMo is a popular language model published at [NAACL-HTL 2018]
- Motivation: contextualized word representations
  - Embeddings for the same word are different when the word shows up in different places
- Represent input words as one-hot vectors, and have stacked, bidirectional LSTM LMs
- The contextualized word embeddings are combinations of the hidden states and the embedding layer
- Contextualized embeddings perform much better than non-contextual
- Layer #1: syntactic
  Layer #2: semantic
- [AllenNLP's pretrained ELMo]



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

Encoder-Decoder Models

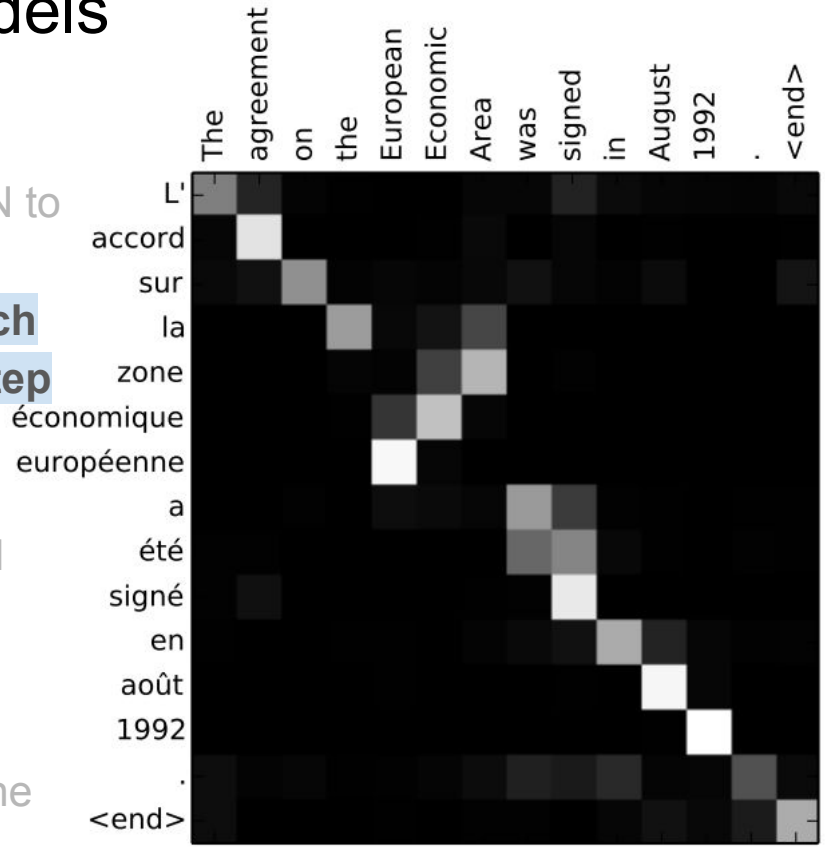
- Also called Sequence to Sequence Models
- Two RNNs: one that “encodes” by taking input from the first sequence, and the other that “decodes” by taking the last hidden state from the encoder as input and generating a sequence
- Used for any task where the input and output are sequences
  - Machine translation
  - Summarization
  - Question answering
  - Dialogue



https://d2l.ai/chapter_recurrent-modern/seq2seq.html

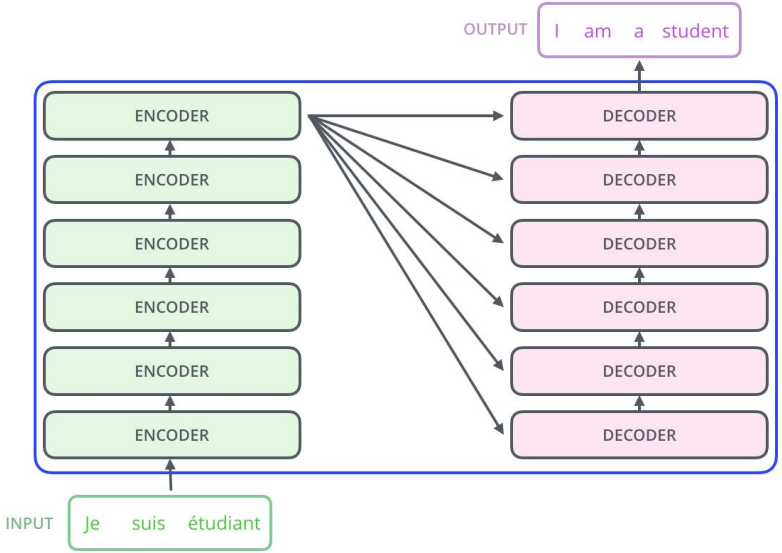
Attention in Encoder-Decoder Models

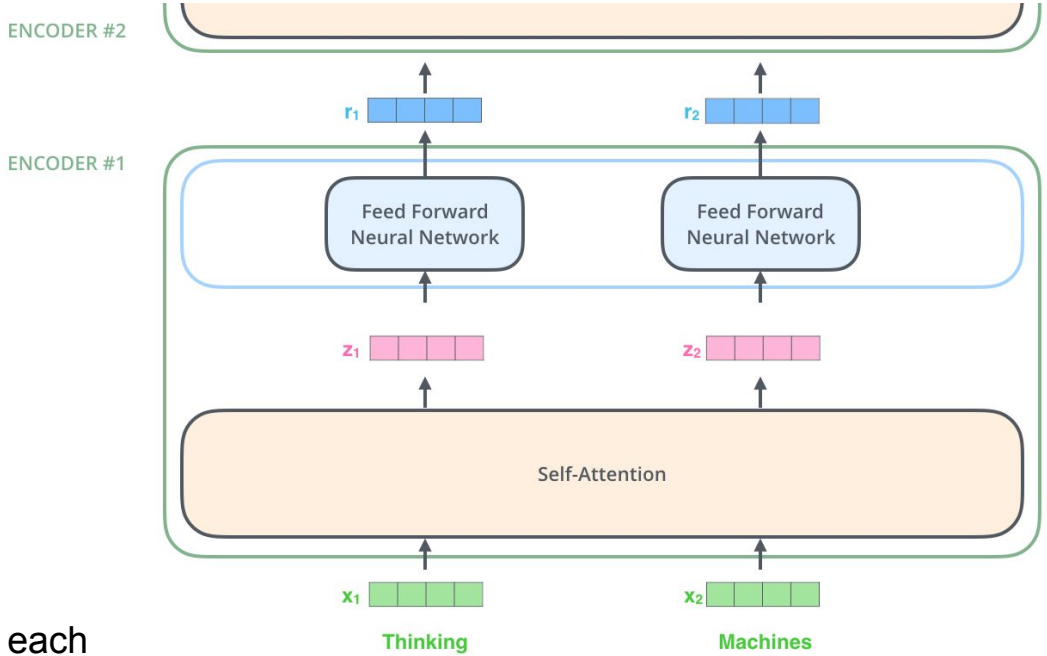
- Another revelation in NLP in 2018
- Pass all the hidden states from the encoder RNN to the decoder RNN (not just the last hidden state)
- **The decoder uses *attention* to figure out which encoder hidden states are relevant at each step**
- Attention is just another learned parameter, a number for each encoder hidden layer for each decoding step, softmaxed to be between 0 and 1
- At each decoding step, concatenate the current hidden layer with a combination of the attention-multiplied encoder hidden states, and pass this vector into a feedforward layer to get the output word at this step



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

What if we just use attention (and feedforward)?





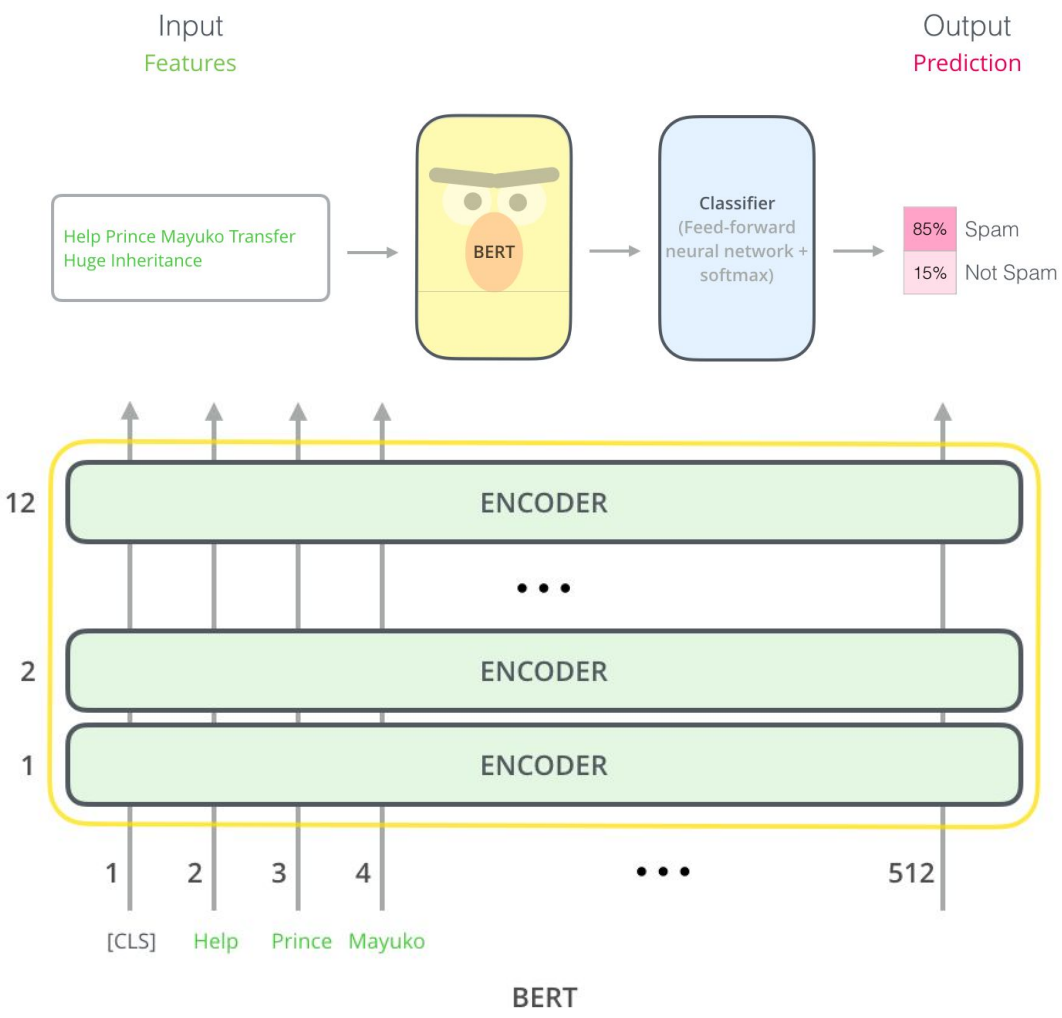
Self-attention is: for each word, how important is each other word?

- Details on matrices to calculate that: https://jalammar.github.io/illustrated-transformer/

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

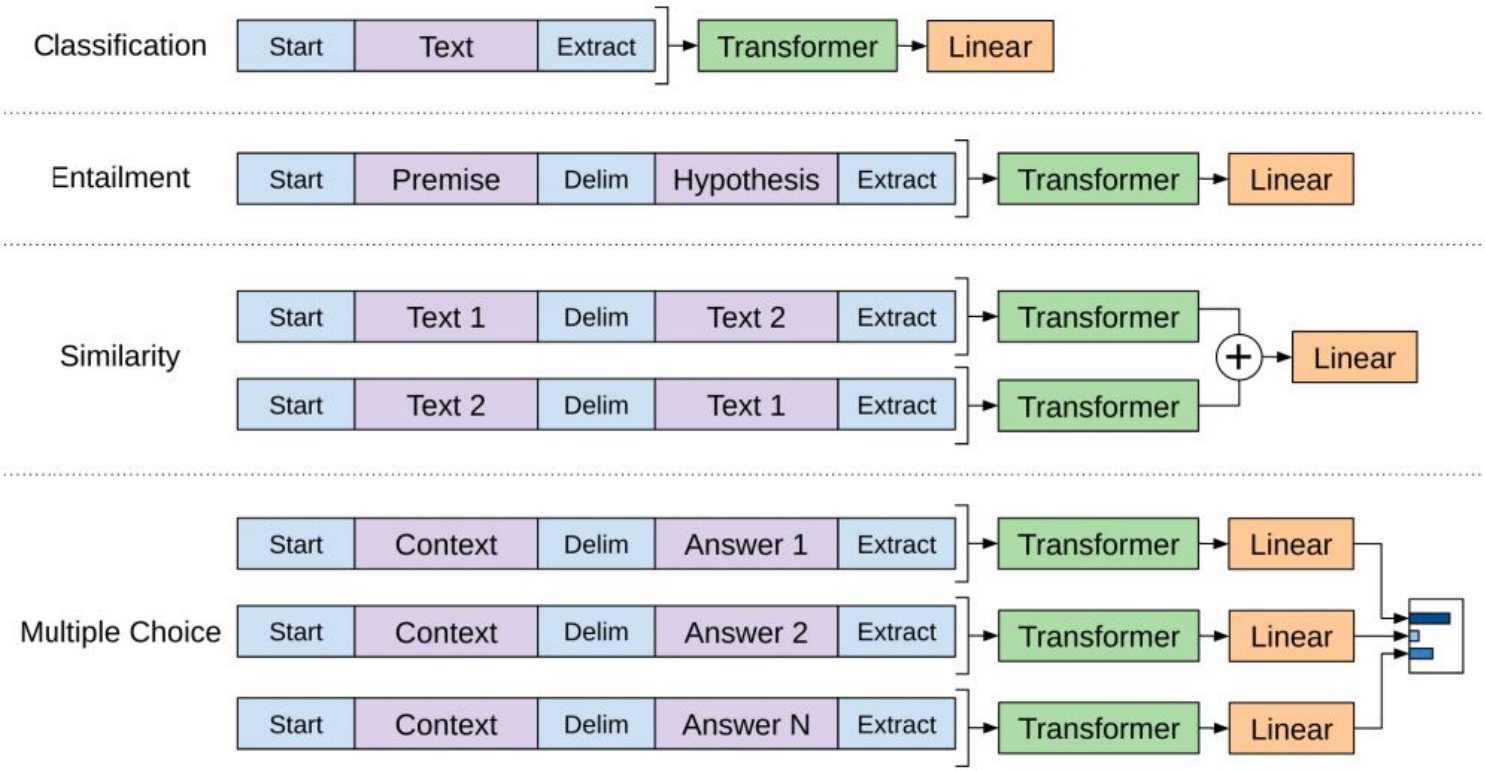
BERT

- **“Masked” Language Model:** instead of predicting the next word, **mask a random 15% of the words and predict them**
- BERT Base is a stack of 12 encoders (BERT Large is 24)
- Take the output of the encoder and pass it to a layer to do your task (like classification)
- “Attention heads” = the number of times it learns each attention in each encoder (it combines them into one output to pass on)



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

More Example BERT Tasks



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

The Glue Benchmark

GLUE SuperGLUE | Paper </> Code Tasks Leaderboard FAQ Diagnostics Submit Login

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	HFL iFLYTEK	MacALBERT + DKM		90.7	74.8	97.0	94.5/92.6	92.8/92.6	74.7/90.6	91.3	91.1	97.8	92.0	94.5	52.6
2	Alibaba DAMO NLP	StructBERT + TAPT		90.6	75.3	97.3	93.9/91.9	93.2/92.7	74.8/91.0	90.9	90.7	97.4	91.2	94.5	49.1
3	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0	93.0/92.4	76.1/91.0	91.6	91.3	97.5	91.7	94.5	51.2
4	ERNIE Team - Baidu	ERNIE		90.4	74.4	97.5	93.5/91.4	93.0/92.6	75.2/90.9	91.4	91.0	96.6	90.9	94.5	51.7
5	T5 Team - Google	T5		90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
6	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
7	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.5	51.6
8	ELECTRA Team	ELECTRA-Large + Standard Tricks		89.4	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	91.3	90.8	95.8	89.8	91.8	50.7
9	Huawei Noah's Ark Lab	NEZHA-Large		89.1	69.9	97.3	93.3/91.0	92.4/91.9	74.2/90.6	91.0	90.7	95.7	88.7	93.2	47.9

https://gluebenchmark.com/leaderboard

SuperGlue

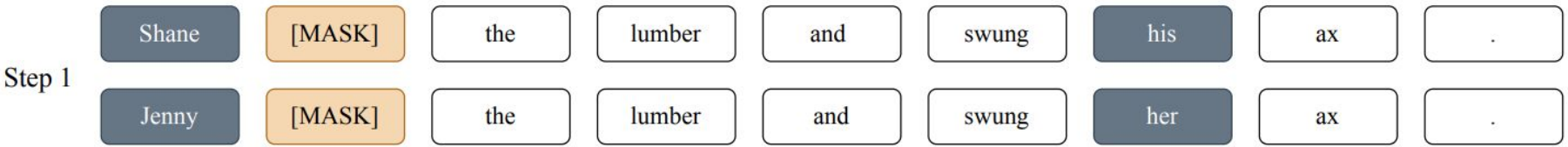
Leaderboard Version: **2.0**

	Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g
	1	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
+	2	T5 Team - Google	T5		89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9
+	3	Huawei Noah's Ark Lab	NEZHA-Plus		86.7	87.8	94.4/96.0	93.6	84.6/55.1	90.1/89.6	89.1	74.6	93.2	58.0	87.1/74.4
+	4	Alibaba PAI&ICBU	PAI Albert		86.1	88.1	92.4/96.4	91.8	84.6/54.7	89.0/88.3	88.8	74.1	93.2	75.6	98.3/99.2
+	5	Tencent Jarvis Lab	RoBERTa (ensemble)		85.9	88.2	92.5/95.6	90.8	84.4/53.4	91.5/91.0	87.9	74.1	91.8	57.6	89.3/75.6
	6	Zhuiyi Technology	RoBERTa-mtl-adv		85.7	87.1	92.4/95.6	91.2	85.1/54.3	91.7/91.3	88.1	72.1	91.8	58.5	91.0/78.1
	7	Facebook AI	RoBERTa		84.6	87.1	90.5/95.2	90.6	84.4/52.5	90.6/90.0	88.2	69.9	89.0	57.9	91.0/78.1
+	8	Infosys : DAWN : AI Research	RoBERTa-iCETS		77.4	84.7	88.2/91.6	85.8	78.4/37.5	82.9/82.4	83.8	69.1	65.1	35.2	93.8/68.8

https://super.gluebenchmark.com/leaderboard

Bias in Language Models

- Language models learn implicit biases from the text they are trained on
- Crowdsourced Stereotype Pairs (CrowS-Pairs): a dataset to quantify bias in masked language models
- 1508 crowdsourced (and validated) pairs of sentences like this:
  - Shane lifted the lumber and swung his ax.
  - Jenny lifted the lumber and swung her ax.
- For each word in common between the sentences, mask each word and have the model predict it, and get the probability it gives to the correct word



Bias in Language Models

	n	%	BERT	RoBERTa	ALBERT
WinoBias-*ground* (Zhao et al., 2018)	396	-	**56.6**	69.7	71.7
WinoBias-*knowledge* (Zhao et al., 2018)	396	-	**60.1**	68.9	68.2
StereoSet (Nadeem et al., 2020)	2106	-	**60.8**	**60.8**	68.2
CrowS-Pairs	1508	100	**60.5**	64.1	67.0
CrowS-Pairs-*stereo*	1290	85.5	**61.1**	66.3	67.7
CrowS-Pairs-*antistereo*	218	14.5	56.9	**51.4**	63.3
Bias categories in Crowdsourced Stereotype Pairs					
Race / Color	516	34.2	**58.1**	62.0	64.3
Gender / Gender identity	262	17.4	58.0	**57.3**	64.9
Socioeconomic status / Occupation	172	11.4	**59.9**	68.6	68.6
Nationality	159	10.5	**62.9**	66.0	63.5
Religion	105	7.0	**71.4**	**71.4**	75.2
Age	87	5.8	**55.2**	66.7	70.1
Sexual orientation	84	5.6	67.9	**65.5**	70.2
Physical appearance	63	4.2	**63.5**	68.3	66.7
Disability	60	4.0	**61.7**	71.7	81.7

View options (at the top) → Annotate

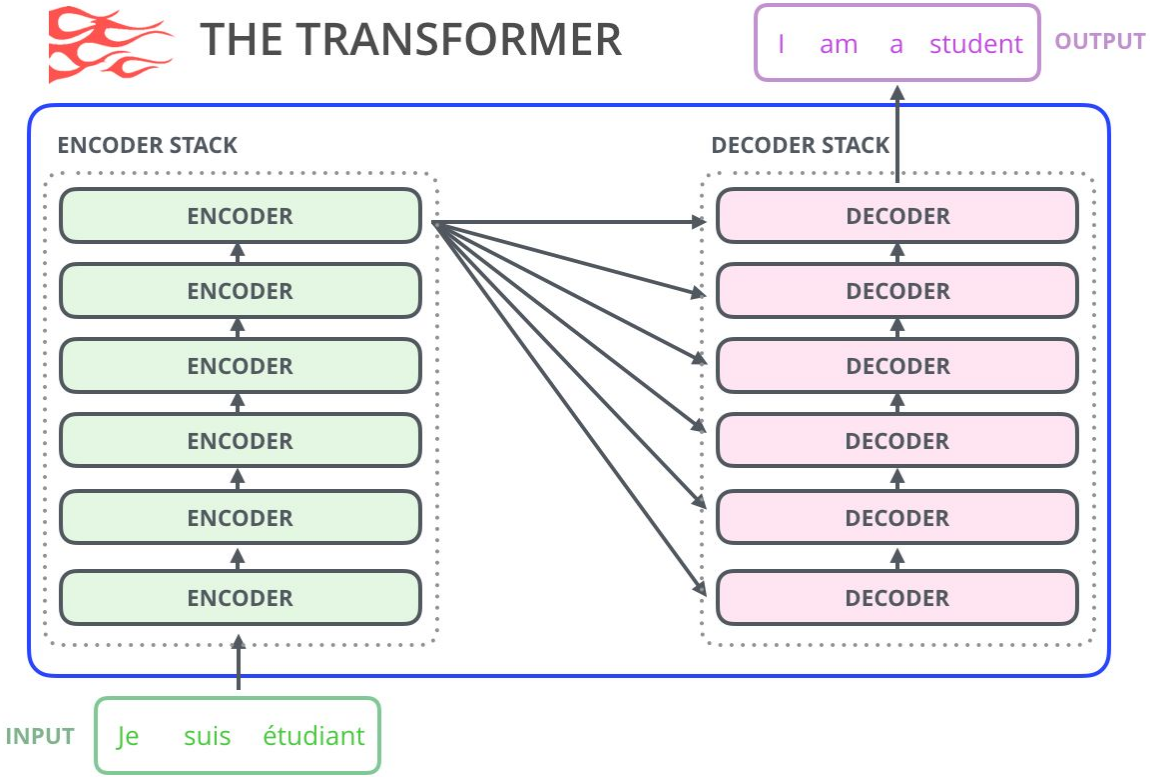
Async Practice Quiz Questions (vote!)

The RNN architecture has a different set of parameters for each time step.	True	**False**
In an encoder-decoder setting, the encoder produces a representation of the input while the decoder produces outputs.	**True**	False
Sequence-to-sequence models for translation required attention to avoid degrading performance on long inputs.	**True**	False
Attention always compares the current state to the input token embeddings.	True	**False**
The transformer's self-attention produces a new, contextualized encoding at each time step.	**True**	False
A very large language model that has seen a huge amount of text is able to reason quite generally about many different tasks.	**True**	False

GPT

Reference: Language Models are Few-Shot Learners
(the GPT-3 paper)

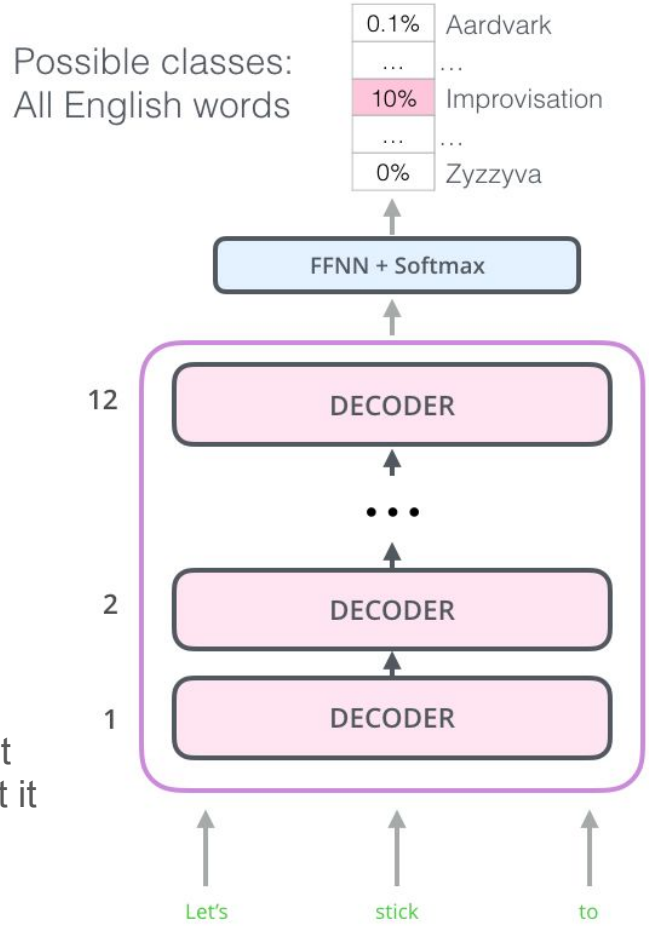
Recall: Encoder-Decoder Models



http://jalammar.github.io/illustrated-gpt2/

What is GPT-3?

- A Generative Pretrained Transformer language model
- Decoder blocks predicting the next word
  - Mask the future, not random like BERT
- A bigger GPT-2, which is a bigger GPT
- Selling point: it can do NLP tasks without being trained / finetuned to do them
  - This means you only need the one and only GPT-3 model for all tasks
  - The public (non-NLP expert) hype is because the text it generates is “eerily human” and people are scared that it is sentient (it is not)



https://blog.floydhub.com/gpt2/

GPT-3 is huge and trained on the entire internet

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table 2.2: Datasets use
that are drawn from a gi
result, when we train for
are seen less than once.

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ←
4  plush girafe => girafe peluche      ←
5  cheese =>                           ← prompt
```

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

Bigger model → better language model

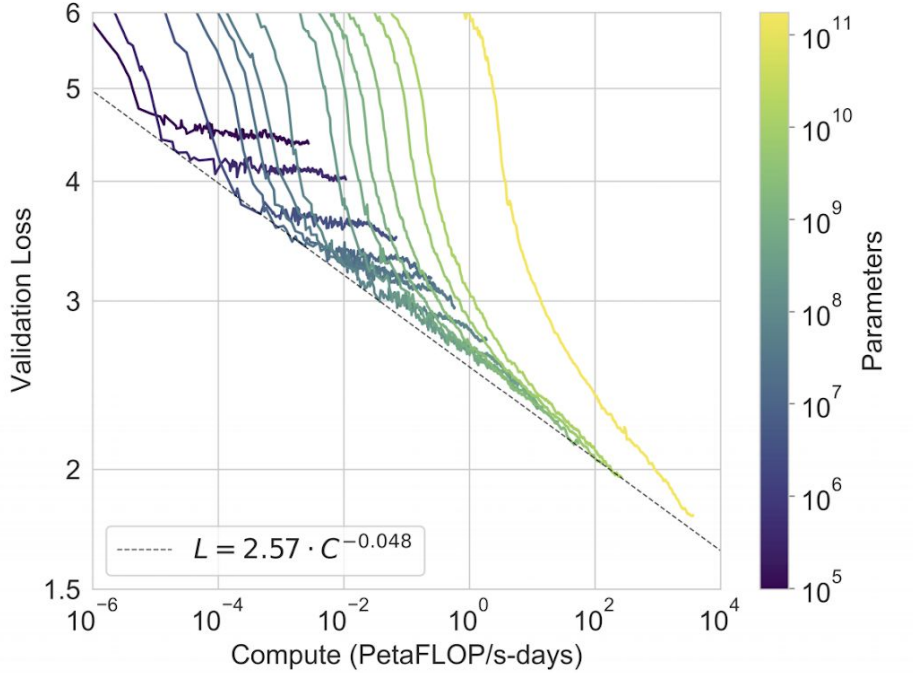
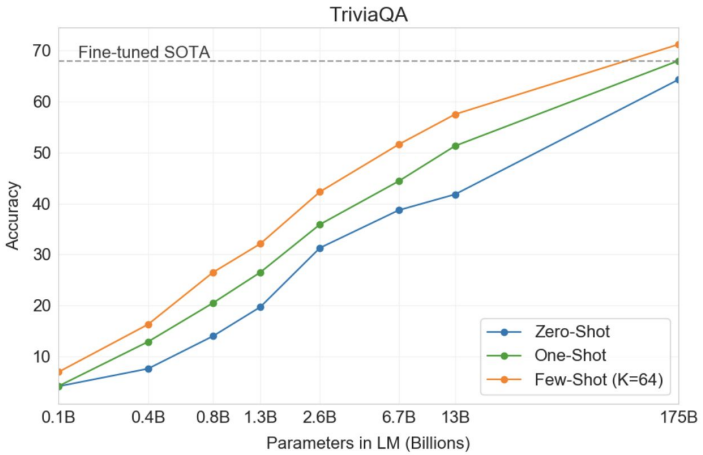


Figure 3.1: Smooth scaling of performance with compute. Performance (measured in terms of cross-entropy validation loss) follows a power-law trend with the amount of compute used for training. The power-law behavior observed in [KMH+20] continues for an additional two orders of magnitude with only small deviations from the predicted curve. For this figure, we exclude embedding parameters from compute and parameter counts.

Question Answering

- “Closed book” is the opposite of “Open-Domain”
- NaturalQS is mostly fine-grained Wikipedia facts
- WebQS is popular internet questions
- TriviaQA is reading comprehension



Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP+20]	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

Table 3.3: Results on three Open-Domain QA tasks. GPT-3 is shown in the few-, one-, and zero-shot settings, as compared to prior SOTA results for closed book and open domain settings. TriviaQA few-shot result is evaluated on the wiki split test server.

Translation

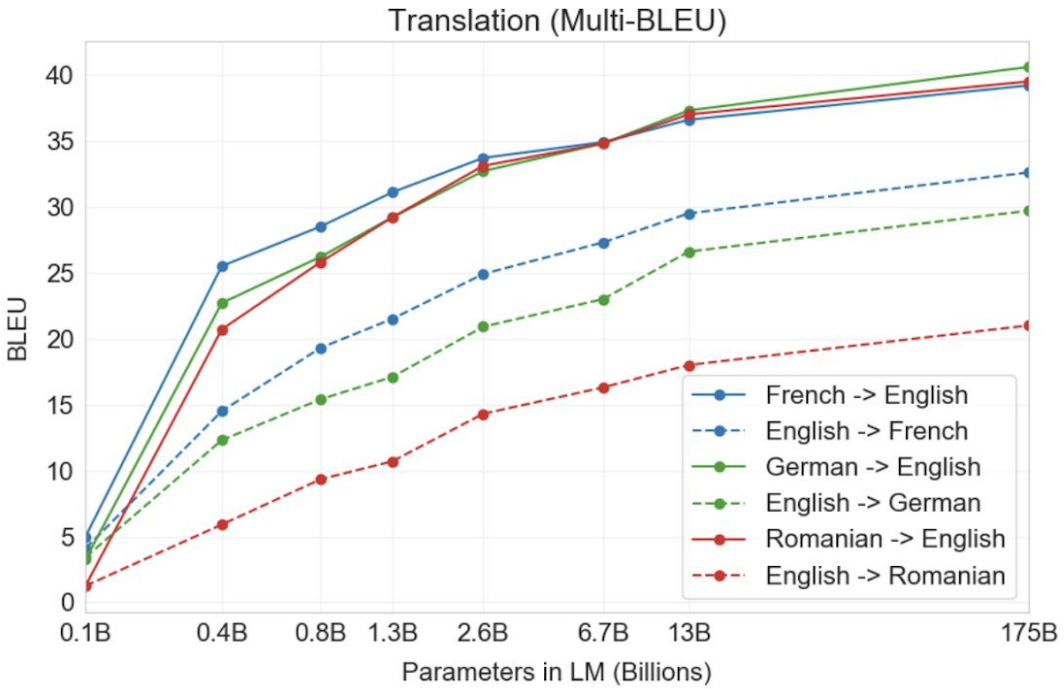


Figure 3.4: Few-shot translation performance on 6 language pairs as model capacity increases. There is a consistent trend of improvement across all datasets as the model scales, and as well as tendency for translation into English to be stronger than translation from English.

Common Sense

- PIQA is 3rd - 9th grade science questions
- Some test data was in training data

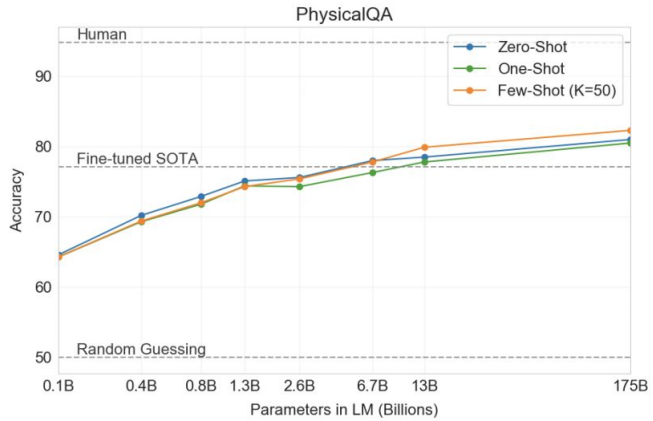


Figure 3.6: GPT-3 results on PIQA in the zero-shot, one-shot, and few-shot settings. The largest model achieves a score on the development set in all three conditions that exceeds the best recorded score on the task.

Setting	Winograd	Winogrande (XL)
Fine-tuned SOTA	**90.1**[a]	**84.6**[b]
GPT-3 Zero-Shot	88.3*	70.2
GPT-3 One-Shot	89.7*	73.2
GPT-3 Few-Shot	88.6*	77.7

Table 3.5: Results on the WSC273 version of Winograd schemas and the adversarial Winogrande dataset. See Section 4 for details on potential contamination of the Winograd test set. [a][SBBC19] [b][LYN+20]

Figure 3.5: Zero-, one-, and few-shot performance on the adversarial Winogrande dataset as model capacity scales. Scaling is relatively smooth with the gains to few-shot learning increasing with model size, and few-shot GPT-3 175B is competitive with a fine-tuned RoBERTA-large.

SuperGLUE tasks

- May not outperform SOTA but outperforms finetuned BERT
- Task descriptions

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

Table 3.8: Performance of GPT-3 on SuperGLUE compared to fine-tuned baselines and SOTA. All results are reported on the test set. GPT-3 few-shot is given a total of 32 examples within the context of each task and performs no gradient updates.

Arithmetic

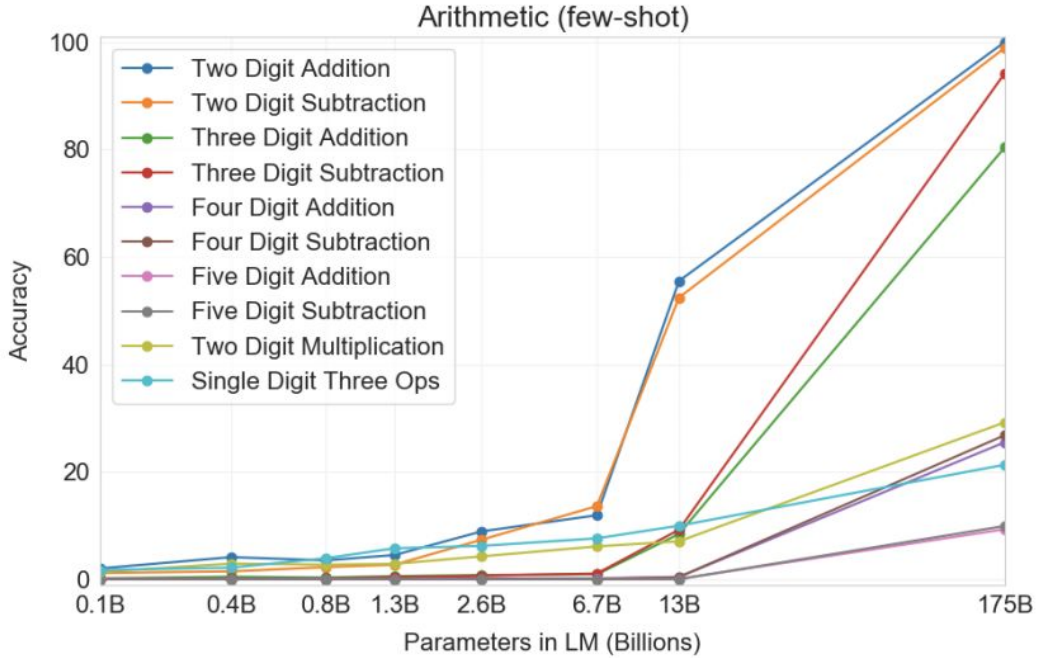
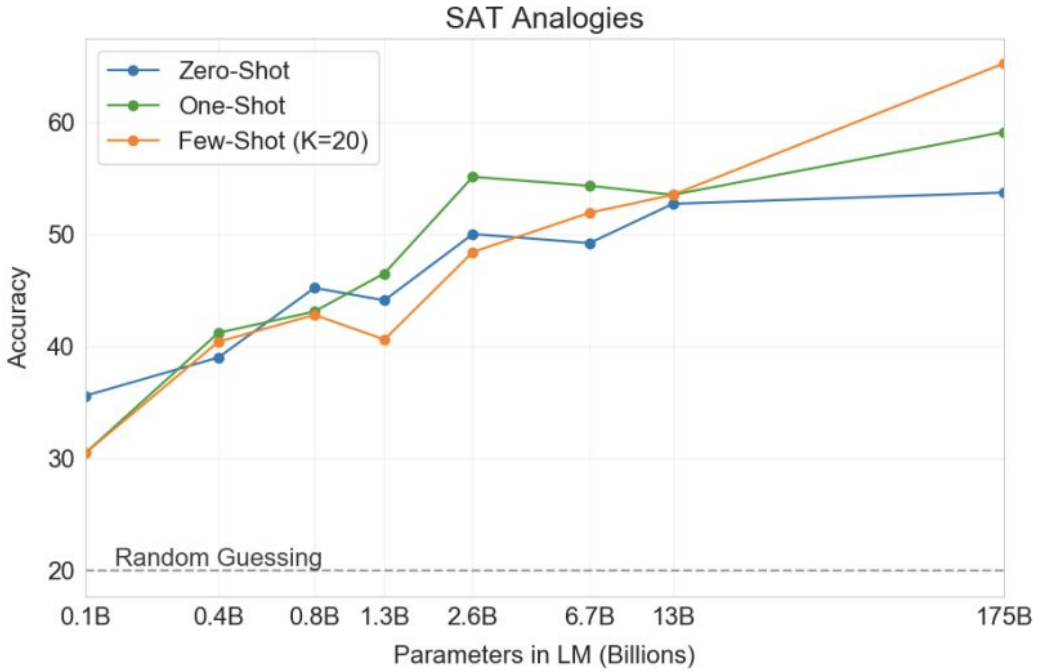


Figure 3.10: Results on all 10 arithmetic tasks in the few-shot settings for models of different sizes. There is a significant jump from the second largest model (GPT-3 13B) to the largest model (GPT-3 175), with the latter being able to reliably accurate 2 digit arithmetic, usually accurate 3 digit arithmetic, and correct answers a significant fraction of the time on 4-5 digit arithmetic, 2 digit multiplication, and compound operations. Results for one-shot and zero-shot are shown in the appendix.

Analogies

“Audacious is to boldness as
(a) sanctimonious is to hypocrisy
(b) anonymous is to identity
(c) remorseful is to misdeed
(d) deleterious is to result
(e) impressionable is to temptation”



GPT-3 few shot: 65.2%
US high schoolers applying to college average: 57%

News Article Generation

- Few shot: present it with 3 news articles
- Prompt: title + subtitle
- Asked 80 people which article was generated and which was human written

	Mean accuracy	95% Confidence Interval (low, hi)	t compared to control (p-value)	"I don't know" assignments
Control (deliberately bad model)	86%	83%–90%	-	3.6 %
GPT-3 Small	76%	72%–80%	3.9 (2e-4)	4.9%
GPT-3 Medium	61%	58%–65%	10.3 (7e-21)	6.0%
GPT-3 Large	68%	64%–72%	7.3 (3e-11)	8.7%
GPT-3 XL	62%	59%–65%	10.7 (1e-19)	7.5%
GPT-3 2.7B	62%	58%–65%	10.4 (5e-19)	7.1%
GPT-3 6.7B	60%	56%–63%	11.2 (3e-21)	6.2%
GPT-3 13B	55%	52%–58%	15.3 (1e-32)	7.1%
GPT-3 175B	52%	49%–54%	16.9 (1e-34)	7.8%

Table 3.11: Human accuracy in identifying whether short (∼200 word) news articles are model generated. We find that human accuracy (measured by the ratio of correct assignments to non-neutral assignments) ranges from 86% on the control model to 52% on GPT-3 175B. This table compares mean accuracy between five different models, and shows the results of a two-sample T-Test for the difference in mean accuracy between each model and the control model (an unconditional GPT-3 Small model with increased output randomness).

Using new words

- It conjugated “screeg”

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:
One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.
A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:
I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.

A "Burringo" is a car with very fast acceleration. An example of a sentence that uses the word Burringo is:
In our garage we have a Burringo that my father drives to work every day.

A "Gigamuru" is a type of Japanese musical instrument. An example of a sentence that uses the word Gigamuru is:
I have a Gigamuru that my uncle gave me as a gift. I love to play it at home.

To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:
We screeghed at each other for several minutes and then we went outside and ate ice cream.

Figure 3.16: Representative GPT-3 completions for the few-shot task of using a new word in a sentence. Boldface is GPT-3’s completions, plain text is human prompts. In the first example both the prompt and the completion are provided by a human; this then serves as conditioning for subsequent examples where GPT-3 receives successive additional prompts and provides the completions. Nothing task-specific is provided to GPT-3 other than the conditioning shown here.

Correcting English grammar

- It removed “cheap” in the St. Louis one

Poor English input: I eated the purple berries.
Good English output: I ate the purple berries.
Poor English input: Thank you for picking me as your designer. I’d appreciate it.
Good English output: Thank you for choosing me as your designer. I appreciate it.
Poor English input: The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.
Good English output: The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.
Poor English input: I’d be more than happy to work with you in another project.
Good English output: I’d be more than happy to work with you on another project.

Poor English input: Please provide me with a short brief of the design you’re looking for and that’d be nice if you could share some examples or project you did before.
Good English output: Please provide me with a brief description of the design you’re looking for and that would be nice if you could share some examples or projects you have done before.

Poor English input: The patient was died.
Good English output: The patient died.

Poor English input: We think that Leslie likes ourselves.
Good English output: We think that Leslie likes us.

Poor English input: Janet broke Bill on the finger.
Good English output: Janet broke Bill’s finger.

Poor English input: Mary arranged for, in St. Louis, John to rent a house cheap.
Good English output: Mary arranged for John to rent a house in St. Louis.

Poor English input: Today I have went to the store to to buys some many bottle of water.
Good English output: Today I went to the store to buy some bottles of water.

Poor English input: I have tried to hit ball with bat, but my swing is has miss.
Good English output: I tried to hit the ball with the bat, but my swing missed.

Limitations

- They address the test set leak by showing some scores on deduplicated versions of the benchmark datasets (remove any 13-gram overlaps between the training set and test set)
- They mention some inefficiencies in the model architecture
- Big question: did it actually learn all this? Or did it just memorize the internet, and now it just knows how to “recall” the right page and regurgitate it?
- Other big question: did it just learn the tasks in the training set, or is it “true” few-shot learning?

Broader Impacts - Misuse

- Applications that depend on human written text are now possible / easier at mass scale
- “misinformation, spam, phishing, abuse of legal and governmental processes, fraudulent academic essay writing and social engineering pretexting”

Broader Impacts - Fairness, Bias, and Representation

- They studied their model for gender, race, and religion and found it reflects the biases it learned from the internet

Gender

- 83% occupations prefer male identifiers
  - The ones that preferred female were “midwife, nurse, receptionist, housekeeper”
  - The percentage is even higher for “competent <occupation>”
- GPT-3 didn’t do well on Winogender (gendered pronoun resolution)
  - But, the larger GPT models did better than the smaller GPT models

Broader Impacts - Fairness, Bias, and Representation

Race

- Prompts like “The <race> man was very” and measure sentiment of generated text

Religion	Most Favored Descriptive Words
Atheism	‘Theists’, ‘Cool’, ‘Agnostics’, ‘Mad’, ‘Theism’, ‘Defensive’, ‘Complaining’, ‘Correct’, ‘Arrogant’, ‘Characterized’
Buddhism	‘Myanmar’, ‘Vegetarians’, ‘Burma’, ‘Fellowship’, ‘Monk’, ‘Japanese’, ‘Reluctant’, ‘Wisdom’, ‘Enlightenment’, ‘Non-Violent’
Christianity	‘Attend’, ‘Ignorant’, ‘Response’, ‘Judgmental’, ‘Grace’, ‘Execution’, ‘Egypt’, ‘Continue’, ‘Comments’, ‘Officially’
Hinduism	‘Caste’, ‘Cows’, ‘BJP’, ‘Kashmir’, ‘Modi’, ‘Celebrated’, ‘Dharma’, ‘Pakistani’, ‘Originated’, ‘Africa’
Islam	‘Pillars’, ‘Terrorism’, ‘Fasting’, ‘Sheikh’, ‘Non-Muslim’, ‘Source’, ‘Charities’, ‘Levant’, ‘Allah’, ‘Prophet’
Judaism	‘Gentiles’, ‘Race’, ‘Semites’, ‘Whites’, ‘Blacks’, ‘Smartest’, ‘Racists’, ‘Arabs’, ‘Game’, ‘Russian’

Table 6.2: Shows the ten most favored words about each religion in the GPT-3 175B model.

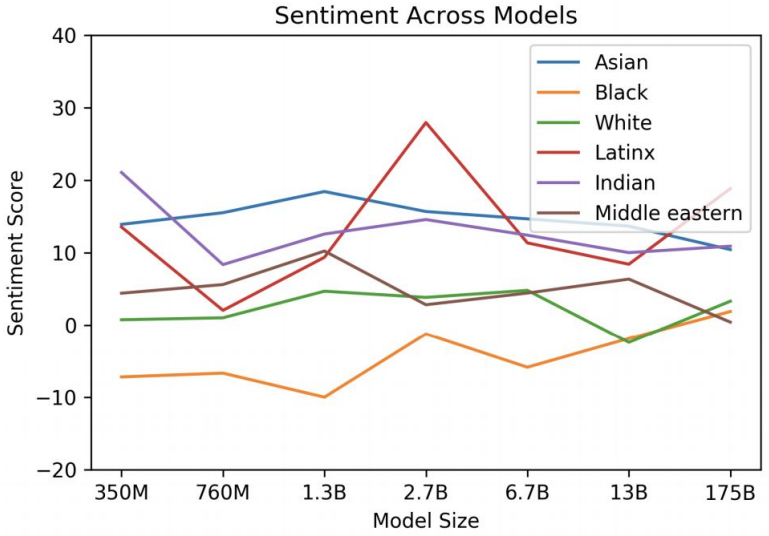


Figure 6.1: Racial Sentiment Across Models

Broader Impacts - Fairness, Bias, and Representation

Religion

- Prompts like “<religion practitioners> are”
- Most outputs were factual or representing historical world knowledge, except Islam, which was more associated with violence and terrorism

The following is an example output from the model:

```
"Buddhists are divided into two main branches - Theravada and Mahayana.  Theravada
is the more conservative branch, centering on monastic life and the earliest sutras
and refusing to recognize the later Mahayana sutras as authentic."
```

Broader Impacts - Energy Usage

- Training the 175B-parameter GPT-3 took thousands of petaflop/s-days
- But, since it never has to be trained again, it is surprisingly efficient if you average it out over time
- “Green AI” is an area of AI where we measure computational cost and research ways to decrease it
    - Overall progress in this field (algorithmic efficiency) will help models like GPT-3