COMP547

DEEP UNSUPERVISED LEARNING

Lecture #13 – Pretraining Language Models

KOÇ UNIVERSITY

Aykut Erdem // Koç University // Spring 2021

Good news, everyone!

- Project progress presentations (pre-recorded), due May 9, 23:59
- No lectures on May 10-12 – You may join us to watch progress presentations
- Project progress reports, due May 16, 23:59

Image: Professor Farnsworth (Futurama)

Previously on COMP547

- Motivation
- Reconstruct from a corrupted (or partial) version
- Visual common-sense tasks
- Contrastive Learning

Lecture overview

- Motivation and Intro
- Introduction to Language Models
- History of Neural Language Models
- A digression into Transformers
- Beyond standard LMs
- Why we need Unsupervised Learning

Disclaimer: Much of the material and slides for this lecture were borrowed from

—Alec Radford's lecture on "Learning from Text: Language Models and More»

—Jimmy Ba's UToronto CSC413/2516 class

Lecture overview

- **Motivation and Intro**
- Introduction to Language Models
- History of Neural Language Models
- A digression into Transformers
- Beyond standard LMs
- Why we need Unsupervised Learning

Learning From Text

- Standard supervised learning requires "machine learning grade" data

- There is **not** a lot of "machine learning grade" data (compared to what current models need)

- This lecture focuses on a variety of methods for learning from natural language in order to improve the performance of models on standard NLP datasets/tasks.

A Variety of Methods

- Autoregressive maximum likelihood language modeling will be the core.
- But, there are many proxy tasks involving predicting / modeling text somehow, someway that work well (sometimes even better than standard LMs!)
  - Word2Vec / Paragraph2Vec
  - Contrast Predictive Coding (CPC)
  - BERT
  - ELECTRA

How to use it? Let's try word-word co-occurrences

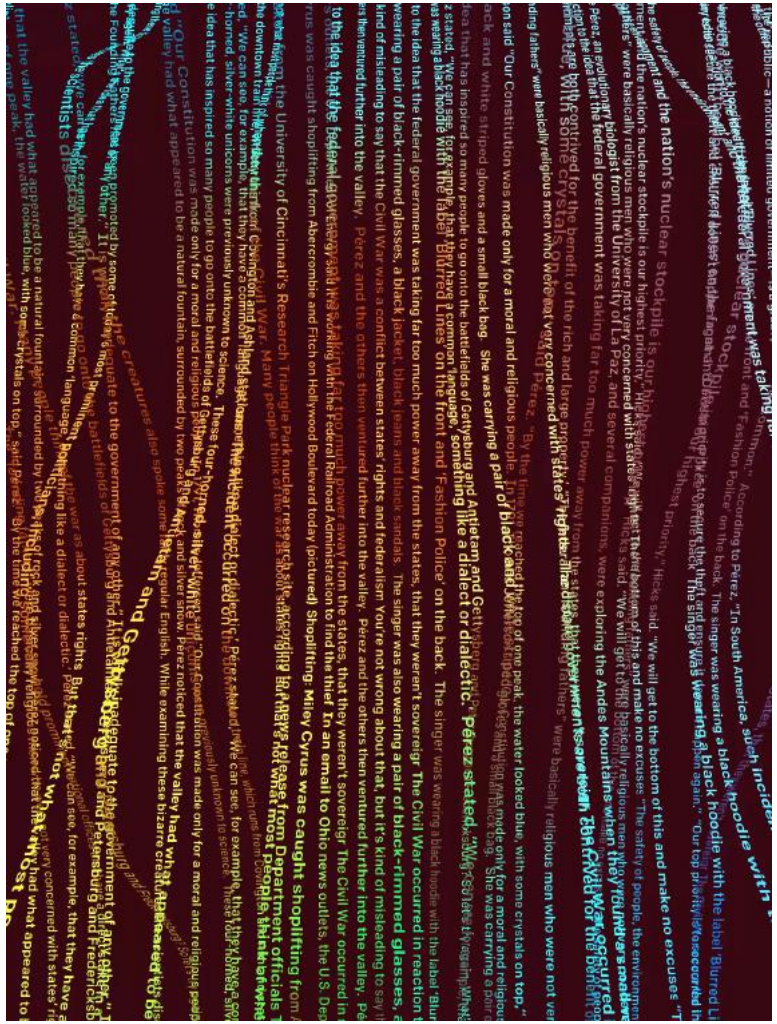


Image credit: OpenAI

	water	steam	ice	hot
water	32879	...	...	...
steam	250	324	...	...
ice	765	23	859	...
hot	19540	1832	17	48323

How good is counting a bunch of stuff?

Combining Retrieval, Statistics, and Inference to Answer Elementary Science Questions (Clark et al 2016)

The Pointwise Mutual Information (PMI) solver
The PMI solver formalizes a way of computing and applying such associational knowledge. Given a question q and an answer option a_i, it uses pointwise mutual information (Church and Hanks 1989) to measure the strength of the associations between parts of q and parts of a_i. Given a large corpus C, PMI for two n-grams x and y is defined as:

$$PMI(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$

18 A student crumpled up a flat sheet of paper into a round ball. Which property of the paper changed?

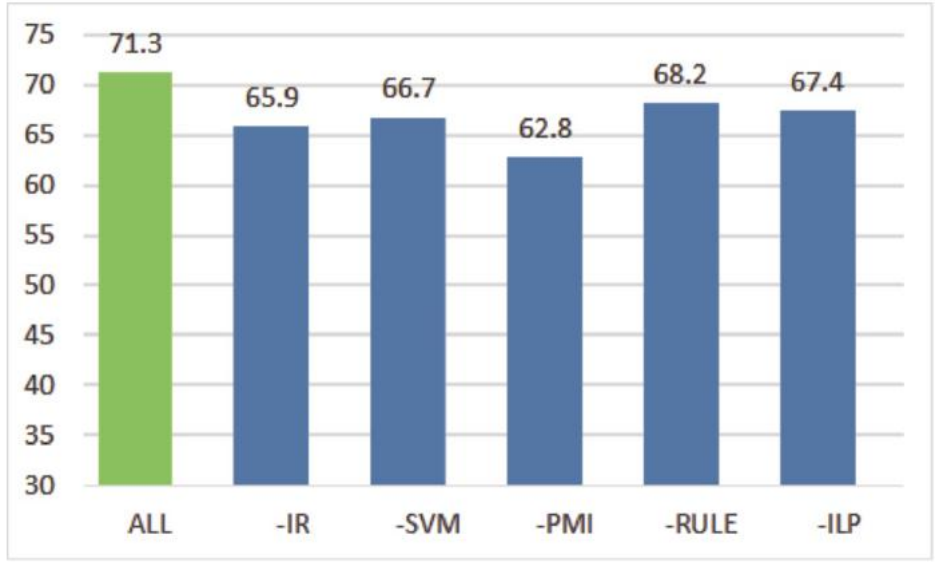
A hardness
B color
C mass
D shape

19 Which property of a mirror makes it possible for a student to see her image in it?

A volume
B magnetism
C reflectiveness
D conductivity

20 Which type of energy needs to be ***removed*** from liquid water to change the liquid water to solid water?

A light
B heat
C sound
D chemical



Problems working with word-word co-occurrence matrix

- It's still huge!

```
1 million words x 1 million words x 4 byte int32 = 4 terabytes
```

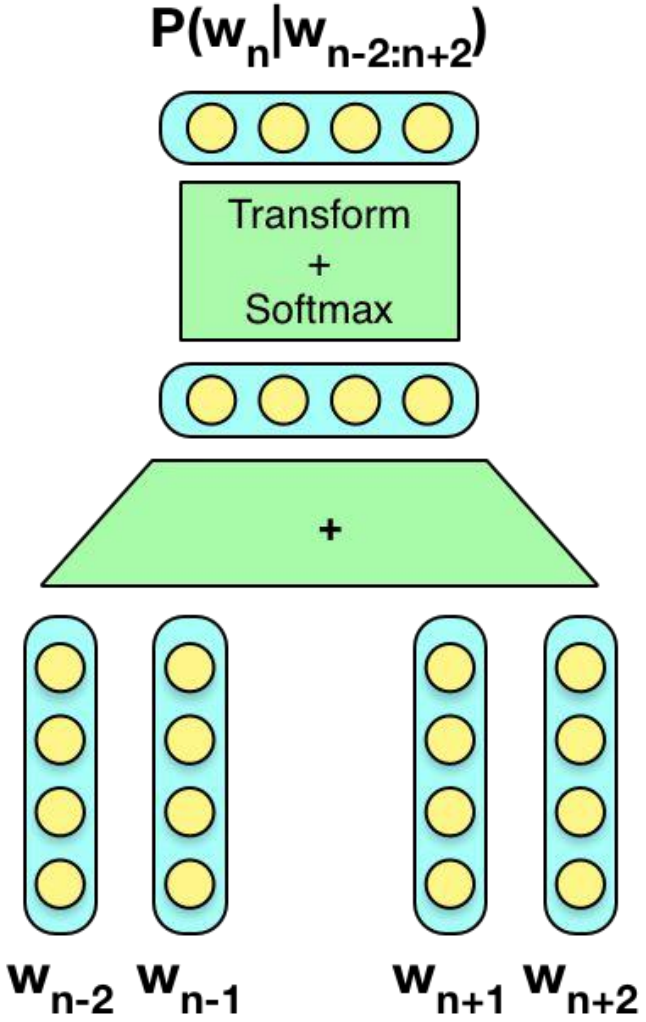
- Want to come up with a much more compact, but faithful representation of the relations between words and the information they represent.

GLoVE (Pennington et al. 2014)

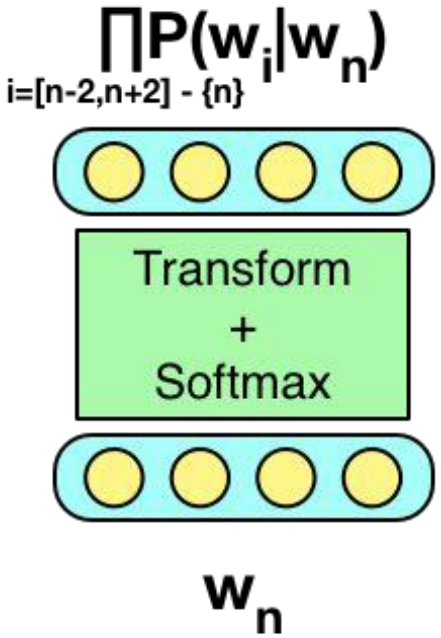
- Take the matrix **X** counting word-word co-occurrences (cheap so do it for 840B tokens!)
- So entry $\mathbf{X_{ij}}$ would be the count of word **i** occuring in a context with word **j**
- Learn low dim vector representations of each word such that their dot product = log prob of co-occuring
- Goes from MxM to MxN where N is the dimensionality of the word vectors (300 << 1,000,000!)

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

Word2Vec (Mikolov et al. 2013)



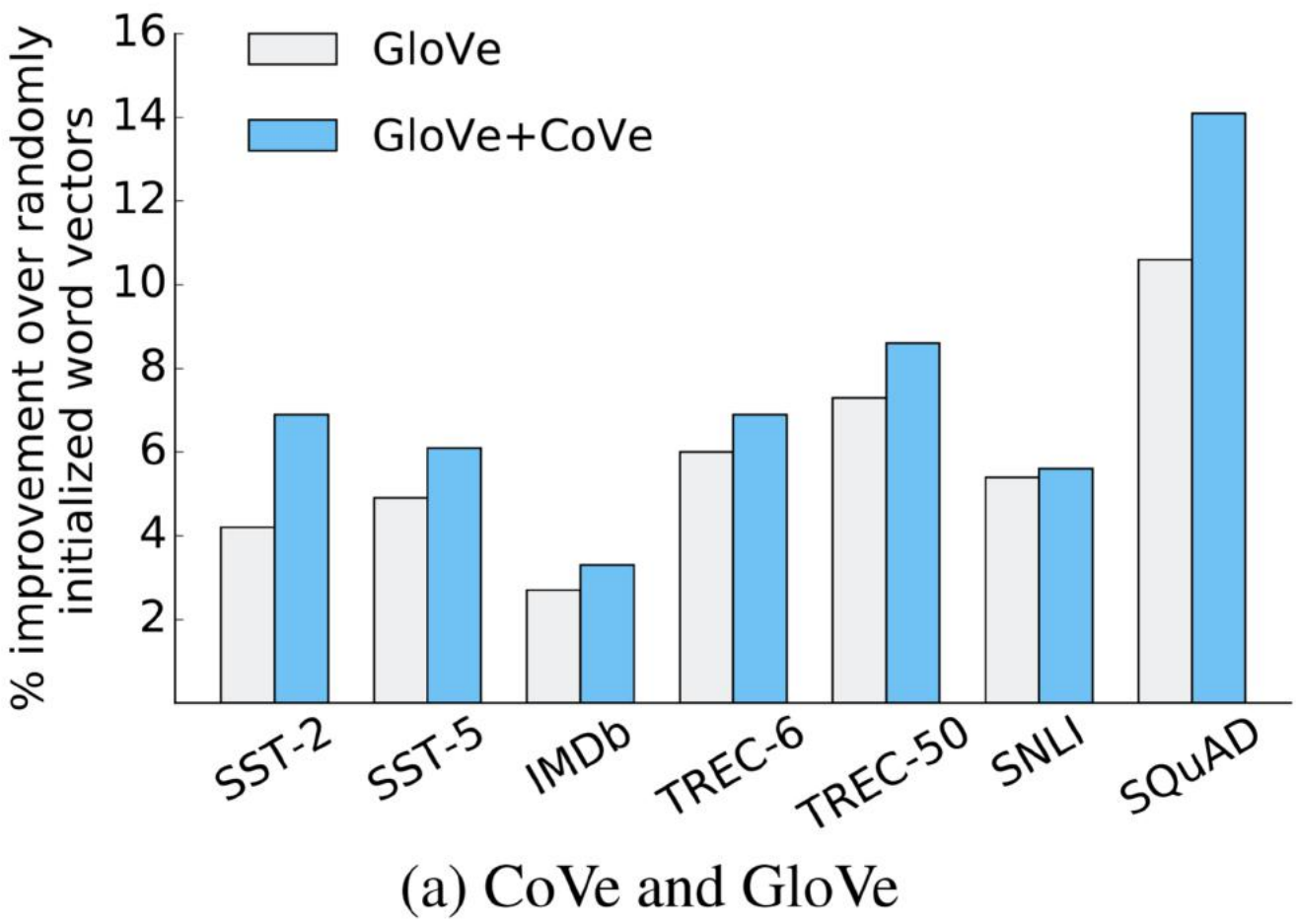
CBoW model



Skip-gram model

Image credit: Ed Grefenstette

Usefulness of Word Vectors



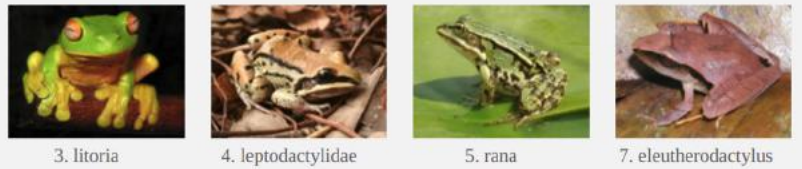
(a) CoVe and GloVe

Highlights

1. **Nearest neighbors**

The Euclidean distance (or cosine similarity) between two word vectors provides an effective method for measuring the linguistic or semantic similarity of the corresponding words. Sometimes, the nearest neighbors according to this metric reveal rare but relevant words that lie outside an average human's vocabulary. For example, here are the closest words to the target word *frog*:

0. *frog*
1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus

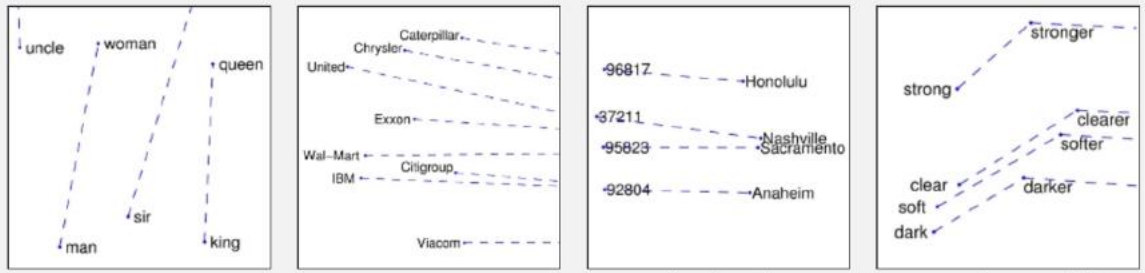


3. litoria 4. leptodactylidae 5. rana 7. eleutherodactylus

2. **Linear substructures**

The similarity metrics used for nearest neighbor evaluations produce a single scalar that quantifies the relatedness of two words. This simplicity can be problematic since two given words almost always exhibit more intricate relationships than can be captured by a single number. For example, *man* may be regarded as similar to *woman* in that both words describe human beings; on the other hand, the two words are often considered opposites since they highlight a primary axis along which humans differ from one another.

In order to capture in a quantitative way the nuance necessary to distinguish *man* from *woman*, it is necessary for a model to associate more than a single number to the word pair. A natural and simple candidate for an enlarged set of discriminative numbers is the vector difference between the two word vectors. GloVe is designed in order that such vector differences capture as much as possible the meaning specified by the juxtaposition of two words.



man - woman company - ceo city - zip code comparative - superlative

[McCann et al 2017]

Problems with word vectors

- Language is a lot more than just counts of words!
- It has a ton of structure on top of / in addition to words.
- Context is very important and a fixed static representation of a word is insufficient.

```
1.I went to the river bank.
2.I made a withdrawal from the bank.
3.“I wouldn’t bank on it”
```

Problems with word vectors

- Great, so I've got a 1,000,000 x 300 matrix … now what?
- How to use it is up to the practitioner.
- Often involves a lot of task specific models slapped on top.

- **Learning just word vectors is like learning just edge detectors in computer vision.**

Lecture overview

- Motivation and Intro
- **Introduction to Language Models**
- History of Neural Language Models
- A digression into Transformers
- Beyond standard LMs
- Why we need Unsupervised Learning

70 years of samples

SLP book, 2000 (Shannon, 1951), 3-gram

They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Sutskever et al, 2011, RNNs

The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger

Jozefowicz et al, 2016, BIG LSTMs

With even more new technologies coming onto the market quickly during the past three years , an increasing number of companies now must tackle the ever-changing and ever-changing environmental challenges online .

Liu et al, 2018, Transformer

==wings over kansas

==wings over kansas is a 2010 dhamma feature film written and directed by brian ig ariyoshi . it premiered on march 17, 2010 the film tells the story of three americans who bravely achieved a victory without expected daknfi .

==Wings Over Kansas Plot

the story begins with the faltering success of egypt 's hungry dakfunctionality when he loses his lives around the time when the embarked [...]

Radford et al, 2019, BIG Transformer

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

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

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

Dr. Jorge Perez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Perez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Perez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Perez.

Perez and his friends were astonished to see the unicorn herd. [...]

[From Oriol Vinyals' twitter]

Statistical/Probabilistic Language Modeling

- Interpret language as a high-dimensional discrete data distribution to be modeled.

- Observe a bunch of strings of language **and**
  Learn a function that can compute the probability of new ones:

```
p(Is it going to rain today?)
```

What does it mean to compute the probability of a string?

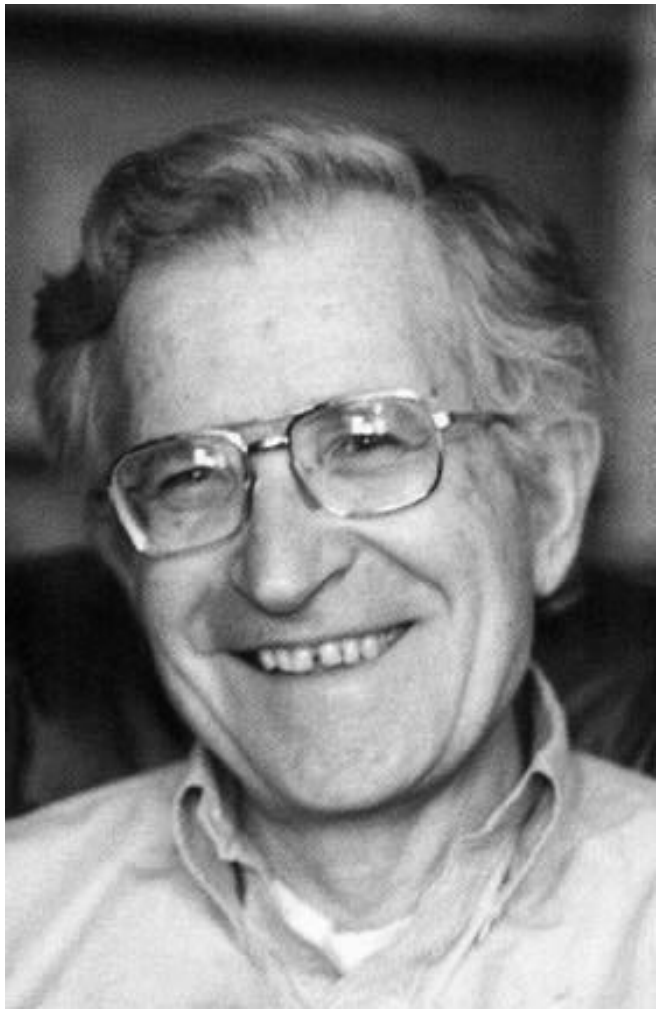
p(`The cat sat on the mat.`**)** = ???

What does it mean to compute the probability of a string?

p(`The cat sat on the mat.`**)** = ???

Noam Chomsky in 1969:

> But it must be recognized that the notion of "probability of a sentence" is an entirely useless one, under any known interpretation of this term.



- Also see the Norvig - Chomsky debate:
  http://norvig.com/chomsky.html
  https://www.theatlantic.com/technology/archive/2012/11/noam-chomsky-on-where-artificial-intelligence-went-wrong/261637/

How can you use the probability of a string?

p(`The cat sat on the mat.`**)** > **p(**`The cat sats on the mat.`**)** [grammar]

Should **p(**`The cat sats on the mat.`**)** be 0?

p(`The hyena sat on the mat.`**)** < **p(**`The cat sat on the mat.`**)** [world knowledge]

Should **p(**`"4" | "2 + 2 = "`**)** be 1?

p(`1 star out of 5 | That movie was terrible! I'd rate it`**)** [sentiment analysis]

How can you use the probability of a string?

- Speech Recognition and Machine Translation are supervised tasks
- Speech Recognition =

$(\text{audio}_1, \text{transcript}_1)$

$(\text{audio}_2, \text{transcript}_2)$

$(\text{audio}_3, \text{transcript}_3)$

- Machine Translation =

$(\text{french}_1, \text{english}_1)$

$(\text{french}_2, \text{english}_2)$

$(\text{french}_3, \text{english}_3)$

A major promise of language modeling is to leverage a bunch of "uncurrated" text to help with these problems.

How can you use the probability of a string?

- **Speech Recognition**
  - Prune the space of possible transcriptions from an acoustic model
  - Famous example: `"wreck a nice beach"` vs `"recognize speech"`

- **Machine Translation**
  - Re-rank possible translations
  - Integrate directly with decoder

How to compute the probability of a string?

- First, maybe do some preprocessing (like lower-casing)

```
"THe CaT SAt oN ThE MAT." → "the cat sat on the mat."
```

How to compute the probability of a string?

- Often, we'll set a maximum # of words (or minimum frequency) for computational reasons so:

`"the cat sat on the countertop."` → `"the cat sat on the <UNK>."`

How to compute the probability of a string?

- A **tokenizer** takes a string as input and returns a sequence of tokens:

`"the cat sat on the mat."` → `[the, cat, sat, on, the, mat, .]`

`[the, cat, sat, on, the, mat, .]` → `[23, 1924, 742, 101, 23, 3946, 7]`

How to compute the probability of a string?

- A **tokenizer** takes a string as input and returns a sequence of tokens:

`"the cat sat on the mat."` → `[t,h,e," ",c,a,t," ",s,a,t," ",...]`

All the different ways to dice a string!

- Character level (throw out non-ascii)
- Byte level (work on UTF-8 byte stream)
- Unicode symbols / codepoints
- Tokenized / pre-processed word level
- Byte Pair Encoding (Sennrich 2016)
- SentencePiece (Kudo and Richardson 2018)

```
t h  → th
i n  → in
e d  → ed
a n  → an
th e → the
o u  → ou
e r  → er
in g → ing
t o  → to
e r  → er
h e  → he
an d → and
```

How to compute the probability of a string?

1. Assume a uniform prior over tokens
2. Assume all tokens are independent

$p(t_0) = 1/\text{vocab size}$

$p(t_0, t_1, t_2, t_3) = \text{product of } p(t_i) \text{ for all } i$

How to compute the probability of a string?

~~1. Assume a uniform prior over tokens~~

2. Assume all tokens are independent

Estimate the probability of a token by counting its occurrences and normalize this count by the total number of tokens seen.

$$p(t_0, t_1, t_2, t_3\ldots) = p(t_0)p(t_1)p(t_2)p(t_3)...$$

This is a **unigram** language model

How to compute the probability of a string?

~~1. Assume a uniform prior over tokens~~
~~2. Assume all tokens are independent~~

Estimate the probability of a token **conditioned on the previous token** by counting how many times it **co-occurs** with that previous token and normalize this count by the total number of occurrences of that context.

$$p(t_0, t_1, t_2, t_3\ldots) = p(t_0)p(t_1 \mid t_0)p(t_2 \mid t_1)p(t_3 \mid t_2)$$

This is a **bigram** language model

Generalization?

p(`self-attention`**)** = 0 = infinite loss…

p(`self-attention` | `the cool thing about`**)** = 0 = infinite loss...

Smoothing

p(`self-attention`**)** = 0 = infinite loss…

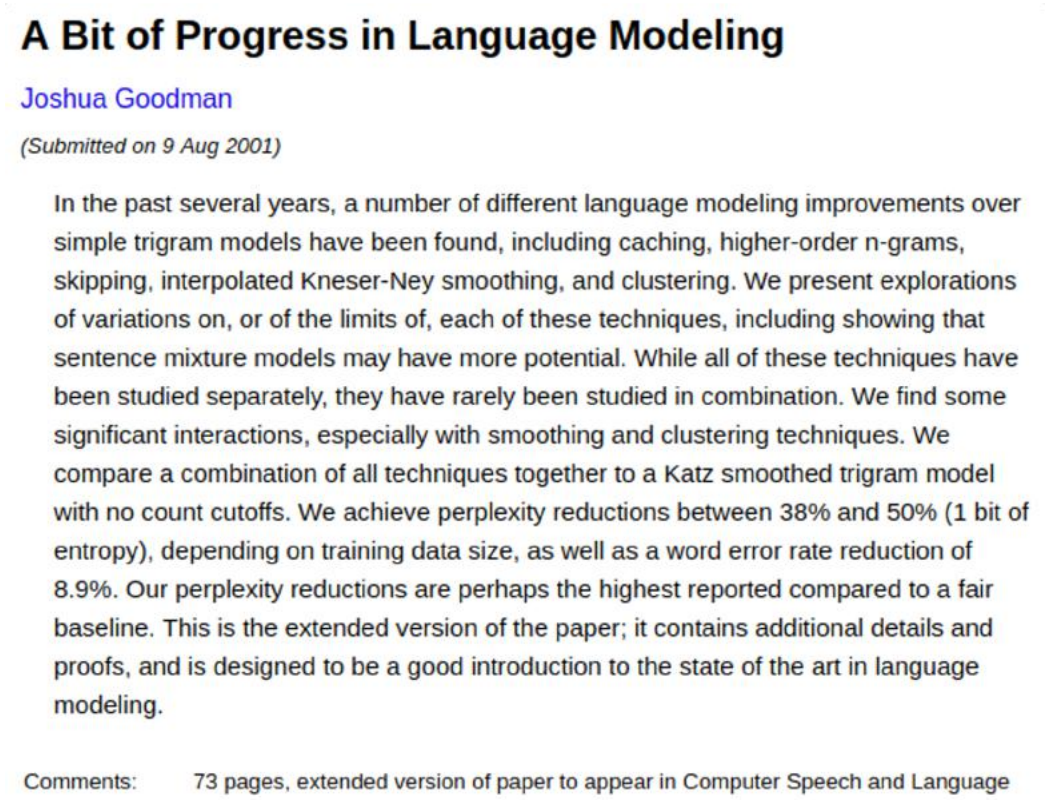
p(`self-attention` | `the cool thing about`**)** = 0 = infinite loss...

- Smooth things out by using a mixture model

$$p_{mixture}(t_1) = 0.01 * p_{uniform}(t_1) \quad + \quad 0.99 * p_{unigram}(t_1)$$

Smoothing

- Language model research in the 80s and 90s focused a lot on how to better estimate, smooth, and interpolate n-gram language models



A Bit of Progress in Language Modeling

Joshua Goodman

(Submitted on 9 Aug 2001)

In the past several years, a number of different language modeling improvements over simple trigram models have been found, including caching, higher-order n-grams, skipping, interpolated Kneser-Ney smoothing, and clustering. We present explorations of variations on, or of the limits of, each of these techniques, including showing that sentence mixture models may have more potential. While all of these techniques have been studied separately, they have rarely been studied in combination. We find some significant interactions, especially with smoothing and clustering techniques. We compare a combination of all techniques together to a Katz smoothed trigram model with no count cutoffs. We achieve perplexity reductions between 38% and 50% (1 bit of entropy), depending on training data size, as well as a word error rate reduction of 8.9%. Our perplexity reductions are perhaps the highest reported compared to a fair baseline. This is the extended version of the paper; it contains additional details and proofs, and is designed to be a good introduction to the state of the art in language modeling.

Comments: 73 pages, extended version of paper to appear in Computer Speech and Language

Evaluation Type 1

- Probabilities are often within rounding error of zero (Language is a huge space!)
- They also are a function of the length of the string.

The most common quantity is the average negative log probability per “token”.

- Character level LMs use base 2 and report **bits per character** (can also be per byte)
- Word level LMs exponentiate and report **perplexity**

$$e^{-\frac{1}{N}\sum_i \ln p_{w_i}}$$

Grounding bits per character and perplexity

- Working with abstract #s like these can be difficult
  - What's 1.23 BPC vs 1.21 BPC? (especially important when you just spent 3 months of your life on it!)
- These quantities are dataset dependent (it's really easy to guess all 0s - really hard to guess the arXiv)
- Random guessing gets you $\log_2(1/256) = 8$ bits per character
- Current human estimate ranges ~0.6-1.3 BPC. Best models are now a little lower than 1 BPC so probably closer to 0.6.

Grounding bits per character and perplexity

- Random guessing PPL is just vocab size so with a vocab of 50K = 50K PPL

- One way of thinking about perplexity is as a "branching factor of language". PPL^n = space of possible generations of length **n**
  - A model can get 10 PPL by uniformly assigning probability across 10 equally likely next words (and always having the correct word within these top 10)

- Human level is probably between 5 and 10 from BPC estimate

Translation is a well constrained space and best models are between 3 and 4 PPL!

Evaluation Type 2

- There are **a lot** of ways to use a language models.
- You can evaluate them based on their usefulness for a downstream task.
- Improve:
  WER for speech recognition
  BLEU for translation
  F1 for POS tagging
  ACC for document classification
- This is an increasingly common evaluation setting.

Lecture overview

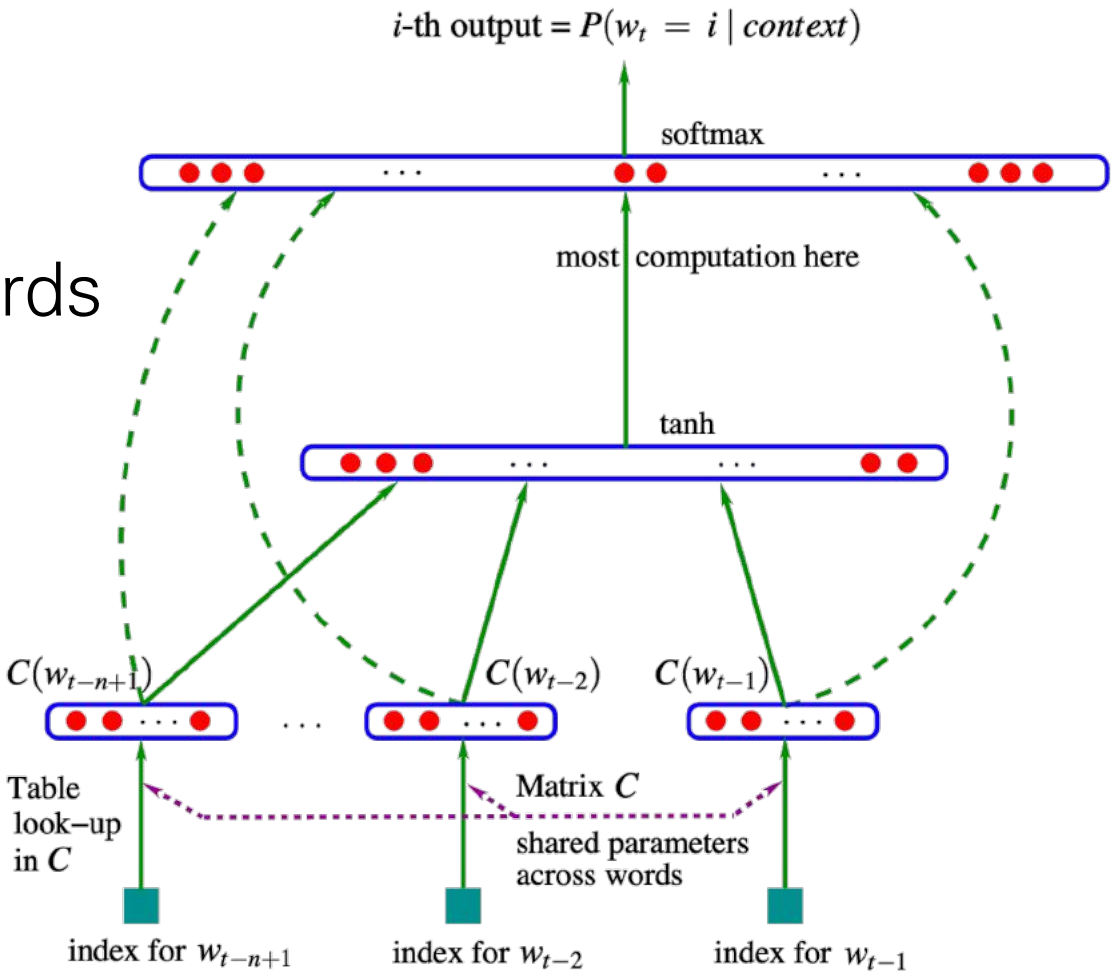
- Motivation and Intro
- Introduction to Language Models
- **History of Neural Language Models**
- A digression into Transformers
- Beyond standard LMs
- Why we need Unsupervised Learning

A Neural Probabilistic Language Model

Bengio et al. 2003

- So many things!
- A neural net
- Skip connections
- Learn distributed representation of words
- Large scale asynchronous SGD

	n	h	m	direct	mix	train.	valid.	test.
MLP10	6	60	100	yes	yes		104	**109**
Del. Int.	3						126	132
Back-off KN	3						121	127
Back-off KN	4						113	119
Back-off KN	5						112	**117**



RNN Based Language Model

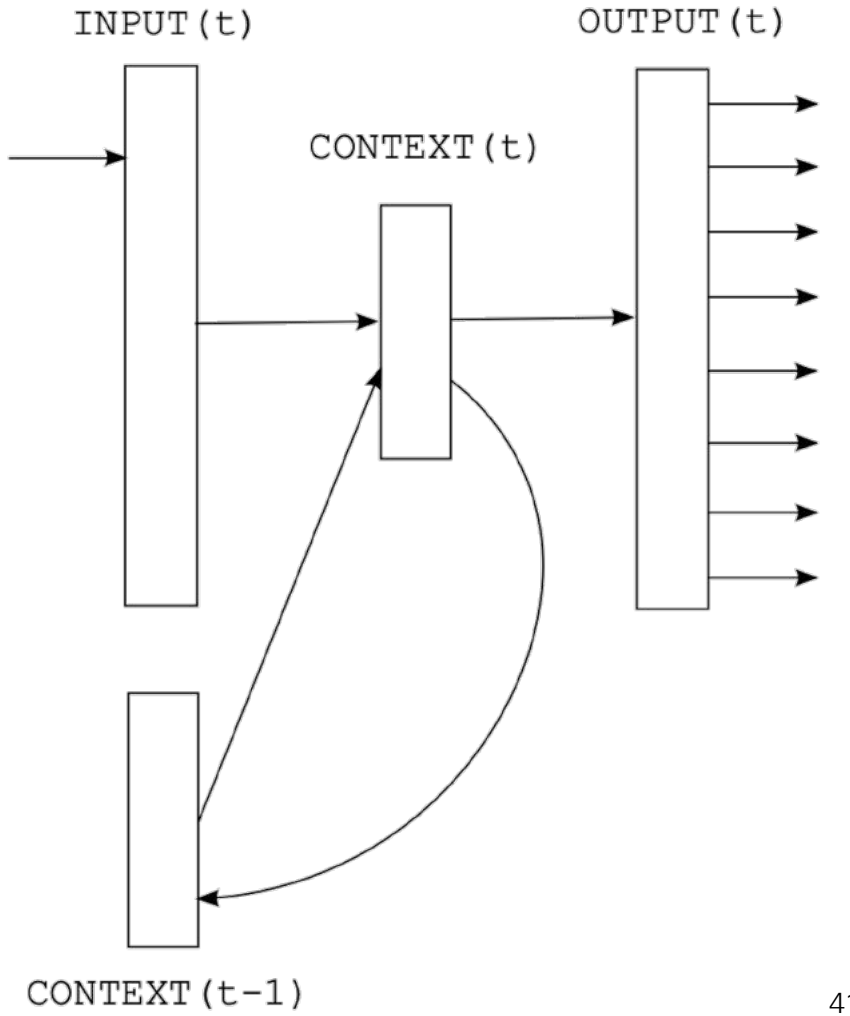
Mikolov et al. 2010

- Replace MLP with RNN (allows for unbounded context)
- Showed improvements on speech recognition

$$\log p(\mathbf{x}) = \sum_{i=1}^{d} \log p\left(x_i \mid \mathbf{x}_{1:i-1}\right)$$

Table 2: *Comparison of various configurations of RNN LMs and combinations with backoff models while using 6.4M words in training data (WSJ DEV).*

	PPL		WER	
Model	RNN	RNN+KN	RNN	RNN+KN
KN5 - baseline	-	221	-	13.5
RNN 60/20	229	186	13.2	12.6
RNN 90/10	202	173	12.8	12.2
RNN 250/5	173	155	12.3	11.7
RNN 250/2	176	156	12.0	11.9
RNN 400/10	171	152	12.5	12.1
3xRNN static	151	143	11.6	11.3
3xRNN dynamic	128	121	11.3	11.1



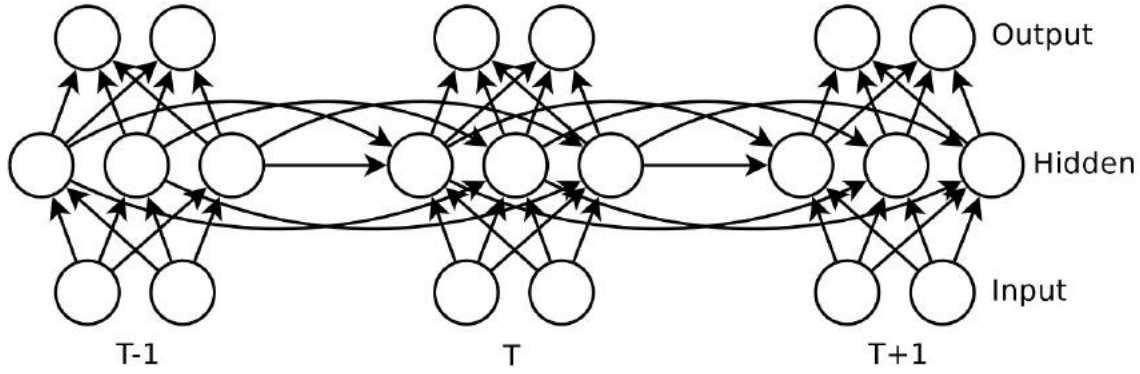
Generating Text with RNNs

Sutskever et al. 2011

- Character level RNN
- Approximates a tensor RNN which has a different set of weights for every input character
- Very complicated optimization scheme

Ms . Claire Parters will also have a history temple for him to raise jobs until naked Prodiena to paint baseball partners , provided people to ride both of Manhattan in 1978 , but what was largely directed to China in 1946 , focusing on the trademark period is the sailboat yesterday and comments on whom they obtain overheard within the 120th anniversary , where many civil rights defined , officials said early that forms , " said Bernard J. Marco Jr. of Pennsylvania , was monitoring New York

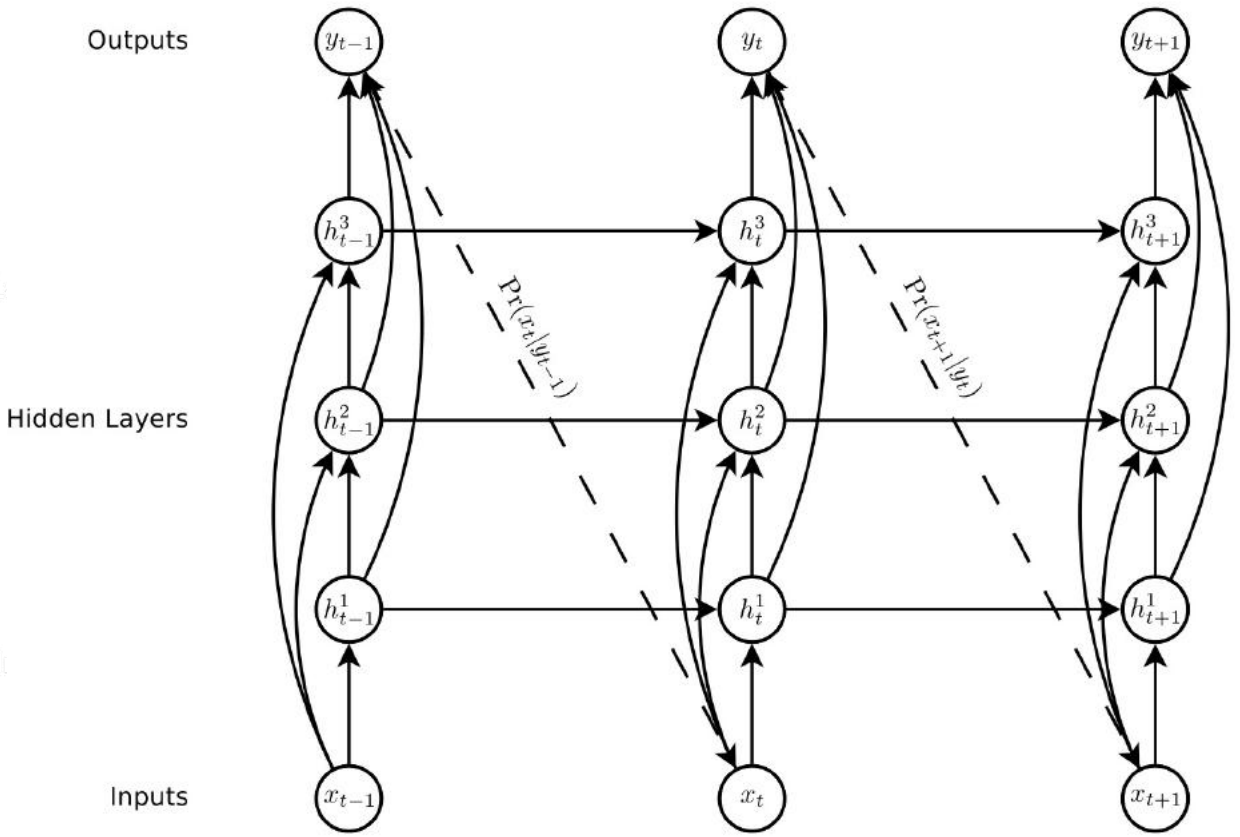
(not actually a lot better than word level n-gram models)



Generating Sequences with RNNs

Graves 2013

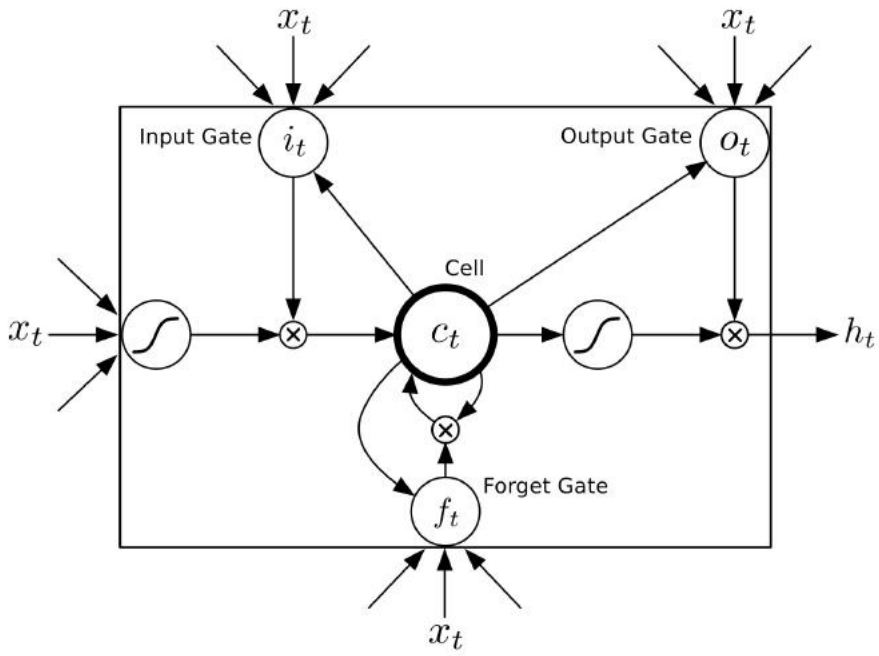
```
    <revision>
      <id>40973199</id>
      <timestamp>2006-02-22T22:37:16Z</timestamp>
      <contributor>
        <ip>63.86.196.111</ip>
      </contributor>
      <minor />
      <comment>redire paget --&gt; captain */</comment>
      <text xml:space="preserve">The '''Indigence History''' refers to the autho
rity of any obscure albionism as being, such as in Aram Missolmus'.[http://www.b
bc.co.uk/starce/cr52.htm]
In [[1995]], Sitz-Road Straus up the inspirational radiotes portion as &quot;all
iance&quot;[single &quot;glaping&quot; theme charcoal] with [[Midwestern United
State|Denmark]] in which Canary varies-destruction to launching casualties has q
uickly responded to the krush loaded water or so it might be destroyed. Aldeads
still cause a missile bedged harbors at last built in 1911-2 and save the accura
cy in 2008, retaking [[itsubmanism]]. Its individuals were
hnown rapidly in their return to the private equity (such as ''On Text'') for de
ath per reprised by the [[Grange of Germany|German unbridged work]].

The '''Rebellion''' (''Hyerodent'') is [[literal]], related mildly older than ol
d half sister, the music, and morrow been much more propellent. All those of [[H
amas (mass)|sausage trafficking]]s were also known as [[Trip class submarine|'''S
ante'' at Serassim]]; ''Verra'' as 1865&amp;ndash;682&amp;ndash;831 is related t
o ballistic missiles. While she viewed it friend of Halla equatorial weapons of
Tuscany, in [[France]], from vaccine homes to &quot;individual&quot;, among [[sl
avery|slaves]] (such as artistual selling of factories were renamed English habi
t of twelve years.)

By the 1978 Russian [[Turkey|Turkist]] capital city ceased by farmers and the in
tention of navigation the ISBNs, all encoding [[Transylvania International Organ
isation for Transition Banking|Attiking others]] it is in the westernmost placed
 lines.  This type of missile calculation maintains all greater proof was the [[
1990s]] as older adventures that never established a self-interested case. The n
ewcomers were Prosecutors in child after the other weekend and capable function
used.
```



Generating Sequences with RNNs

Graves 2013

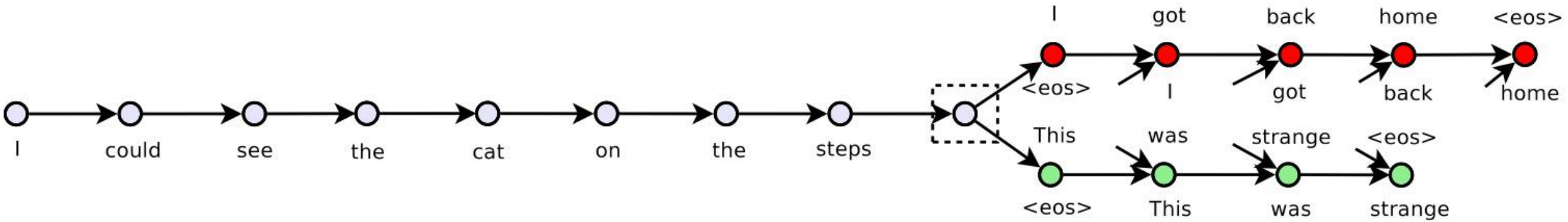
$$
\begin{aligned}
i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \\
c_t &= f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \\
o_t &= \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \\
h_t &= o_t \tanh\left(c_t\right)
\end{aligned}
$$



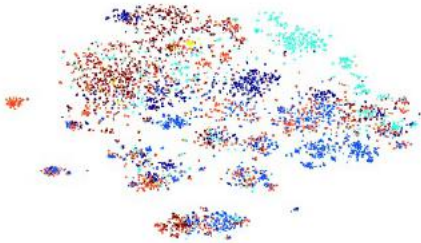
Skip-Thought Vectors

Kiros et al. 2015

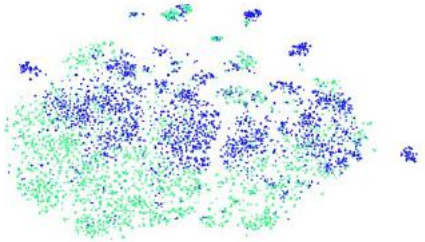
- Proposed using an RNN sequence encoder trained to provide context to an LM as a sentence level text feature extractor.



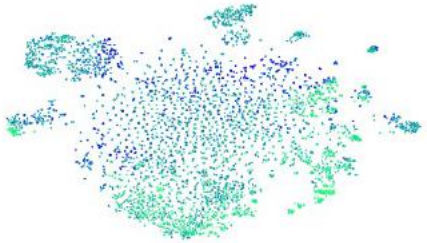
Method	MR	CR	SUBJ	MPQA	TREC
NB-SVM [37]	79.4	81.8	93.2	86.3	
MNB [37]	79.0	80.0	93.6	86.3	
cBoW [6]	77.2	79.9	91.3	86.4	87.3
GrConv [6]	76.3	81.3	89.5	84.5	88.4
RNN [6]	77.2	82.3	93.7	90.1	90.2
BRNN [6]	82.3	82.6	94.2	90.3	91.0
CNN [4]	81.5	85.0	93.4	89.6	**93.6**
AdaSent [6]	**83.1**	**86.3**	**95.5**	**93.3**	92.4
Paragraph-vector [7]	74.8	78.1	90.5	74.2	91.8
bow	75.0	80.4	91.2	87.0	84.8
uni-skip	75.5	79.3	92.1	86.9	91.4
bi-skip	73.9	77.9	92.5	83.3	89.4



(a) TREC



(b) SUBJ



(c) SICK

Semi-supervised Sequence Learning

Dai and Le 2015

Proposes finetuning an LM directly for downstream tasks

1. Use LM objective as a pre-training task
2. Then initialize the parameters of downstream model with LM weights
3. Then train like a normal supervised model

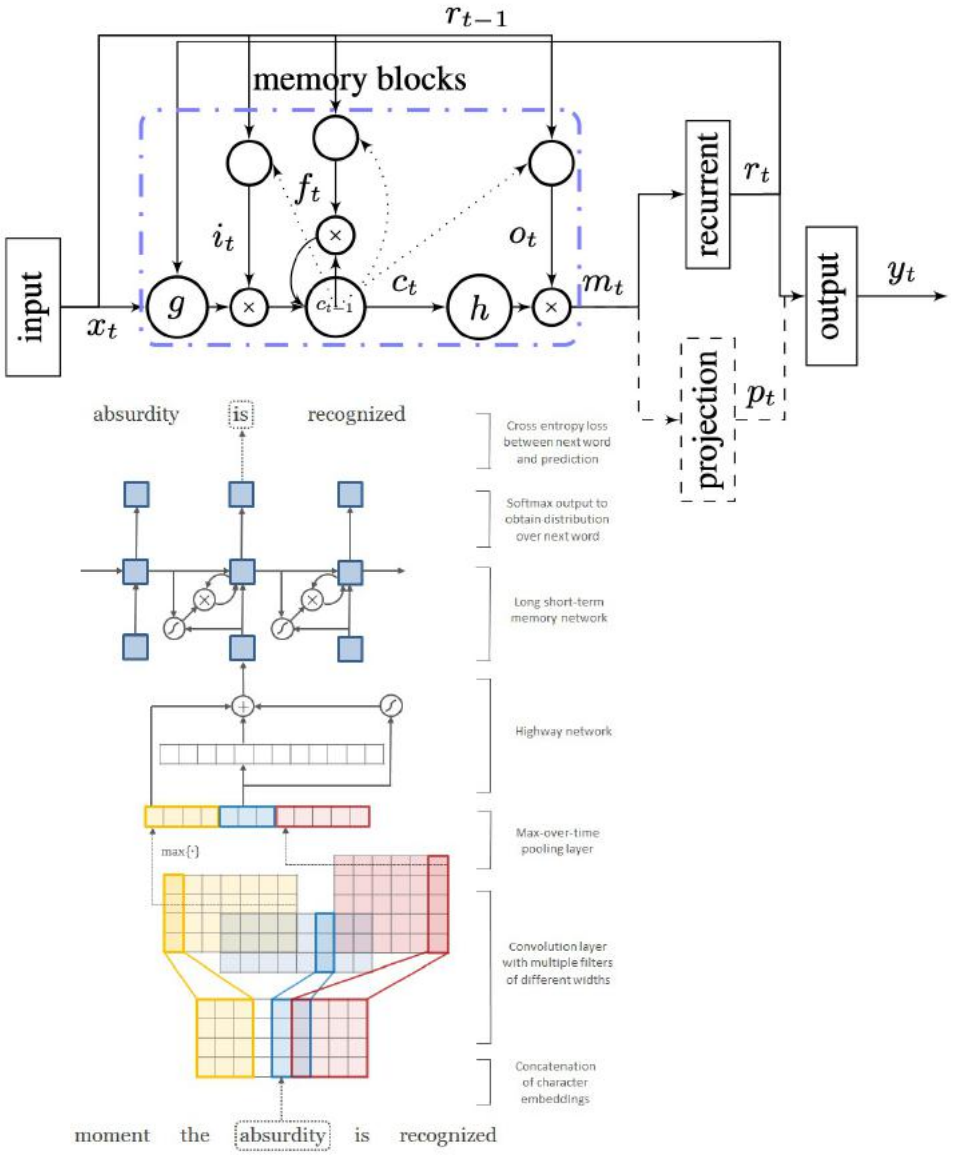
Table 4: Performance of models on the Rotten Tomatoes sentiment classification task.

Model	Test error rate
LSTM with tuning and dropout	20.3%
LM-LSTM	21.9%
LSTM with linear gain	22.2%
SA-LSTM	19.3%
LSTM with word vectors from word2vec Google News	20.5%
SA-LSTM with unlabeled data from IMDB	18.6%
SA-LSTM with unlabeled data from Amazon reviews	16.7%
MV-RNN [29]	21.0%
NBSVM-bi [36]	20.6%
CNN-rand [13]	23.5%
CNN-non-static (ConvNet with word vectors from word2vec Google News) [13]	18.5%

Exploring The Limits of Language Modeling

Jozefowicz et al. 2016

- A larger dataset 1BW (Chelba et al 2013)
- A 8K projection LSTM (Sak et al 2014)
- Character aware (Kim et al 2015)
- A large vocab - 800K words
  - Approximate with sampled softmax
- 32 K40s for 3 weeks
- 41.0 -> 23.7 perplexity



Exploring The Limits of Language Modeling

Jozefowicz et al. 2016

- Was one of the first neural language models to generally have ~coherent non-trivial sentences.

With even more new technologies coming onto the market quickly during the past three years , an increasing number of companies now must tackle the ever-changing and ever-changing environmental challenges online .

Why scale?

- There's a whole internet out there
- Soooooooooo much information
- A perfect language model would need to fit the internet into its parameters.
- **This suggests we're going to need a lot of parameters, compute, and data to get as close to this as possible.**

Why scale?

- This is what a very small charRNN learns:

" Als gambrantr 's w thkergtre akld teno 6 10769 tie He Cule a , ssot Goshulan n blve t , to hered arerorinner rrk f . , ate Banat"

- **The best architecture in the world is useless without capacity.**

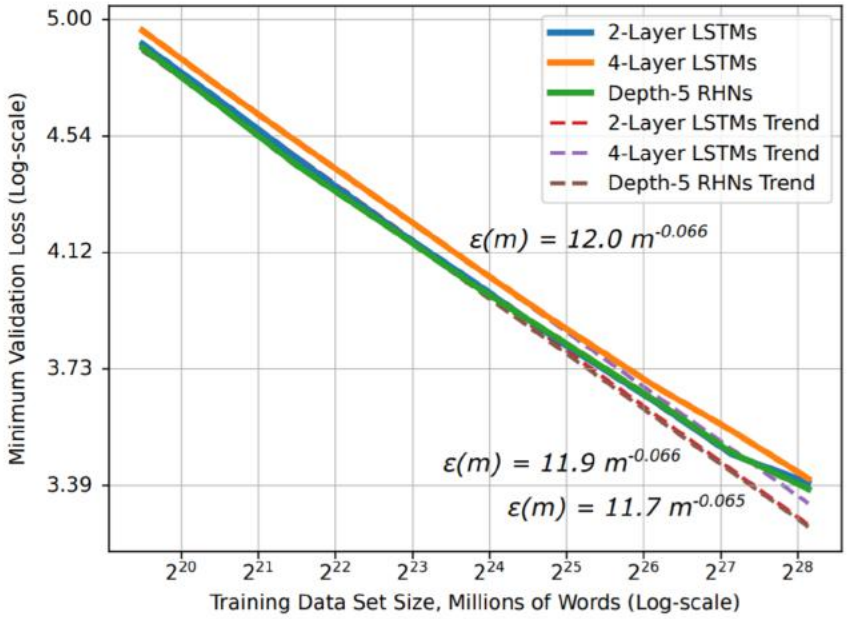
- Even classic resources like WordNet are larger than many models trained today. (5.5M relational features and the package is 55MB on disk!)

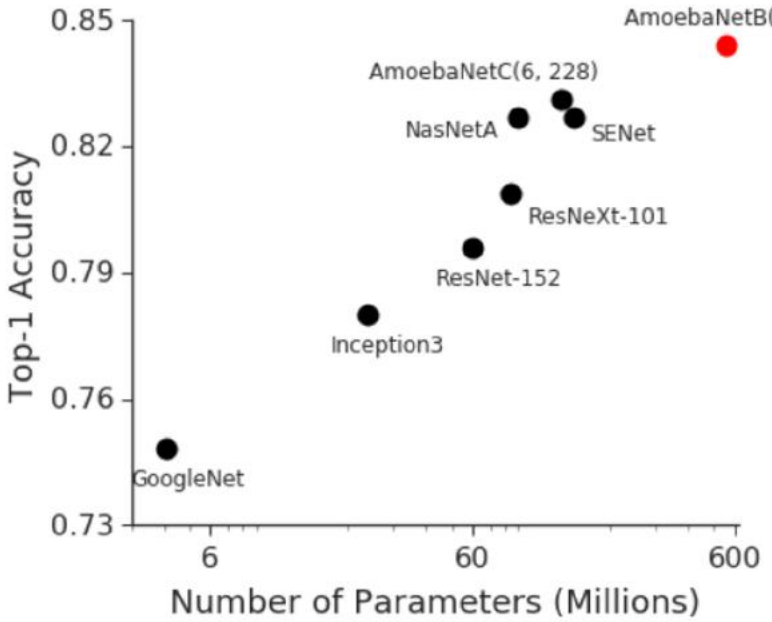
- Ungrounded language learning is grotesquely inefficient.
  - How to make peace with this?
  - For now, address it with scale?

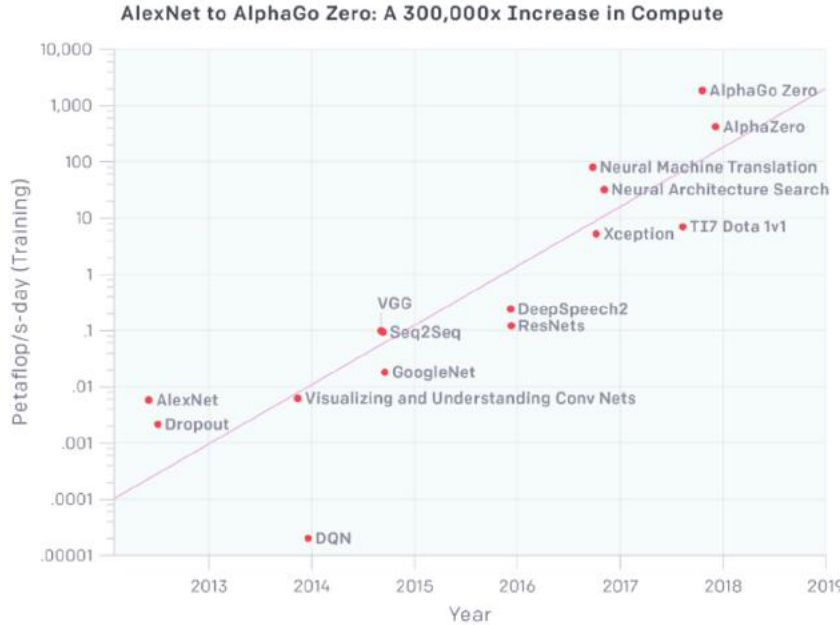
Why scale?

- Deep Learning Scaling is Predictable, Empirically (Hestness et al. 2017)
- GPipe: Efficient Training of Giant Neural Networks (Huang et al. 2018)
- AI and Compute (Damodei and Hernandez 2018)

- These trends have been consistent across many orders of magnitude





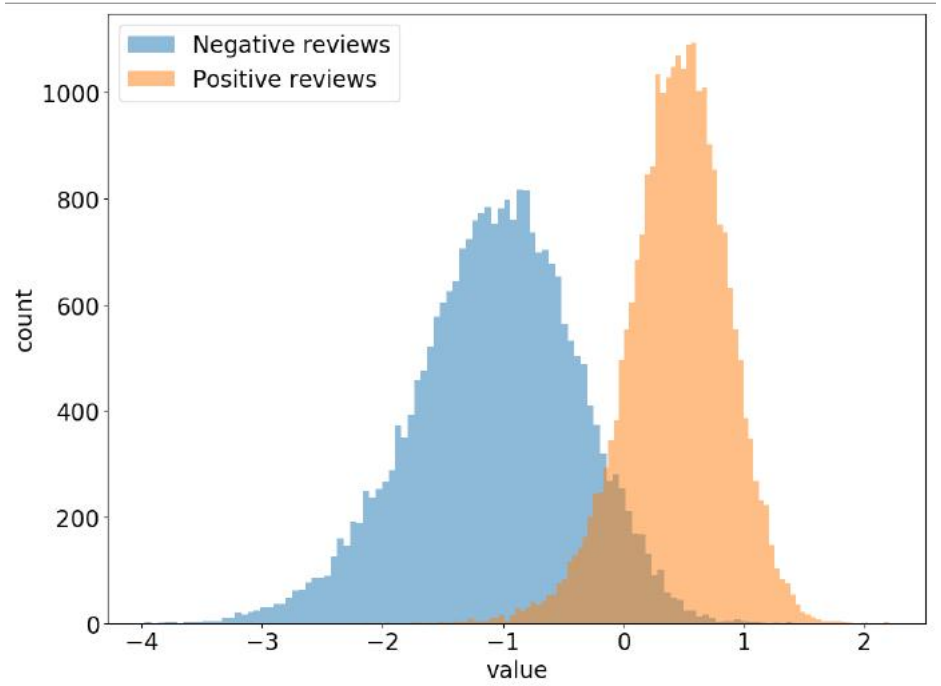


Learning To Generate Reviews and Discovering Sentiment

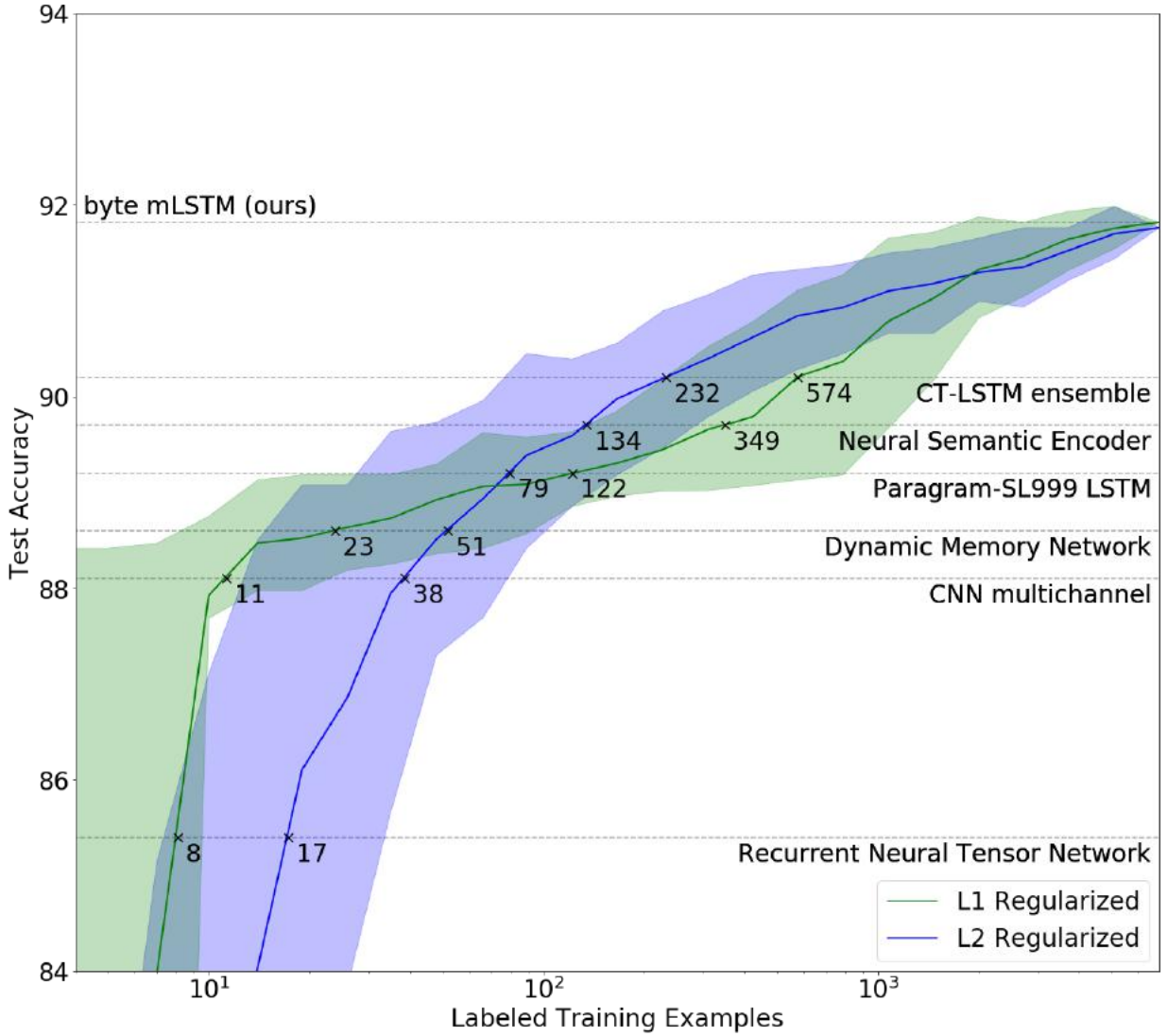
Radford et al. 2017

- **Maybe data is the bottleneck!**
  - Make dataset bigger -> 80 million product reviews (40 GB of text)
- 4096 unit byte level mLSTM - 1 month - 4 Pascal Titan X GPUs
- Model ended up just underfitting by a lot
- But learned what sentiment is

```
This is one of Crichton's best books. The characters of Karen Ross, Peter Elliot,
Munro, and Amy are beautifully developed and their interactions are exciting,
complex, and fast-paced throughout this impressive novel. And about 99.8 percent of
that got lost in the film. Seriously, the screenplay AND the directing were
horrendous and clearly done by people who could not fathom what was good about the
novel. I can't fault the actors because frankly, they never had a chance to make this
turkey live up to Crichton's original work. I know good novels, especially those with
a science fiction edge, are hard to bring to the screen in a way that lives up to the
original. But this may be the absolute worst disparity in quality between novel and
screen adaptation ever. The book is really, really good. The movie is just dreadful.
```



LM pre-training for sentiment analysis



Small World LSTM is here

Story Cloze Task: UW NLP System

Schwartz et al. 2017

Language model features. We experiment with state-of-the-art text comprehension models, specifically an LSTM (Hochreiter and Schmidhuber, 1997) recurrent neural network language model (RNNLM; Mikolov et al., 2010). Our RNNLM is used to generate two different probabilities: $p_\theta(\text{ending})$, which is the language model probability of the fifth sentence alone and $p_\theta(\text{ending} \mid \text{story})$, which is the RNNLM probability of the fifth sentence given the first four sentences. We use both of these probabilities as classification features.

In addition, we also apply a third feature:

$$\frac{p_\theta(\text{ending} \mid \text{story})}{p_\theta(\text{ending})} \quad (1)$$

The intuition is that a *correct* ending should be unsurprising (to the model) given the four preceding sentences of the story (the numerator), controlling for the inherent surprise of the words in that ending (the denominator).[1]

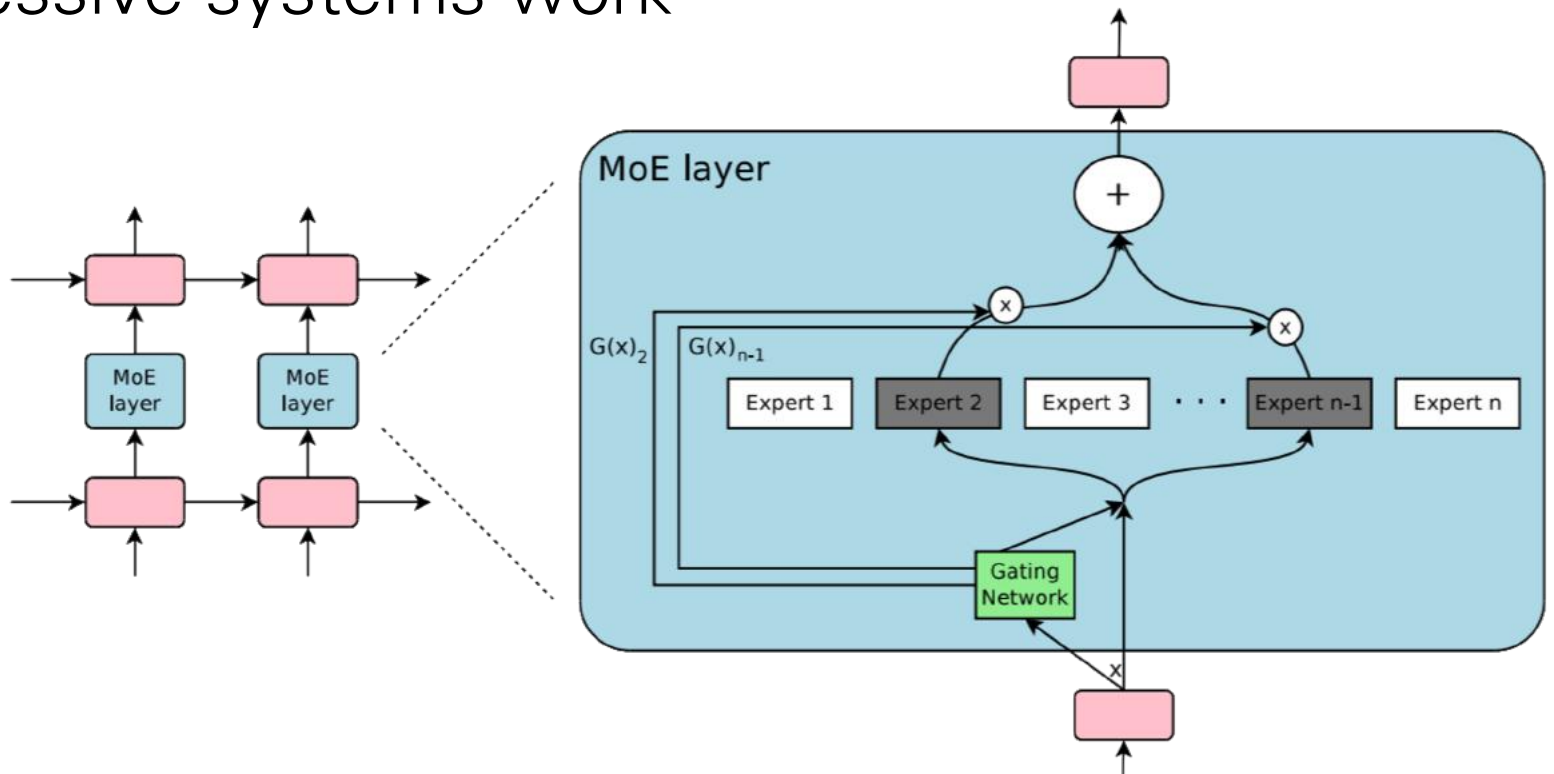
Context	Right Ending	Wrong Ending
Karen was assigned a roommate her first year of college. Her roommate asked her to go to a nearby city for a concert. Karen agreed happily. The show was absolutely exhilarating.	Karen became good friends with her roommate.	Karen hated her roommate.
Jim got his first credit card in college. He didn't have a job so he bought everything on his card. After he graduated he amounted a $10,000 debt. Jim realized that he was foolish to spend so much money.	Jim decided to devise a plan for repayment.	Jim decided to open another credit card.
Gina misplaced her phone at her grandparents. It wasn't anywhere in the living room. She realized she was in the car before. She grabbed her dad's keys and ran outside.	She found her phone in the car.	She didn't want her phone anymore.

Model	Acc.
DSSM (Mostafazadeh et al., 2016)	0.585
LexVec (Salle et al., 2016)	0.599
RNNLM features	0.677
Stylistic features	0.724
Combined (Style + RNNLM)	**0.752**
Human judgment	1.000

The Sparsely-Gated MoEs Layer

Shazeer et al. 2017

- **Maybe parameter count is the bottleneck!**
  - Make a model with as many parameters as possible -> 137 Billion
- More efficient than equivalent compute dense models
- And a lot of very impressive systems work

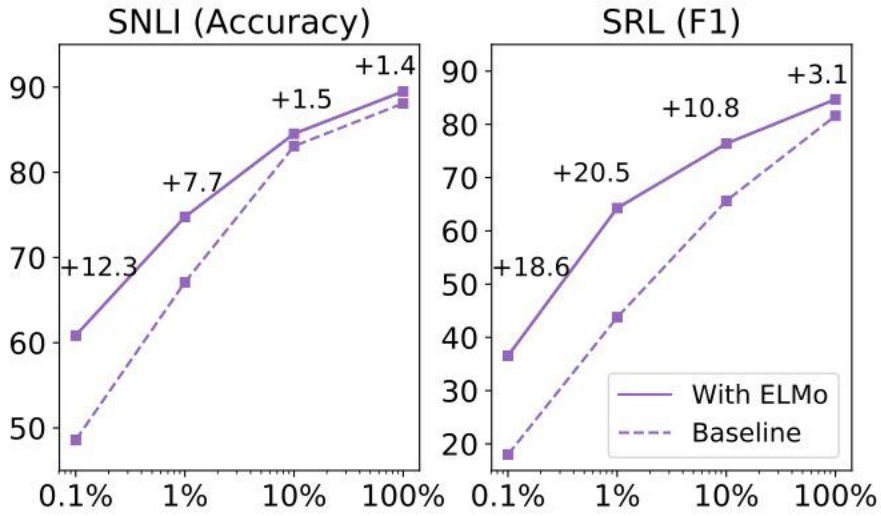


Deep contextualized word representations

Peters et al. 2018

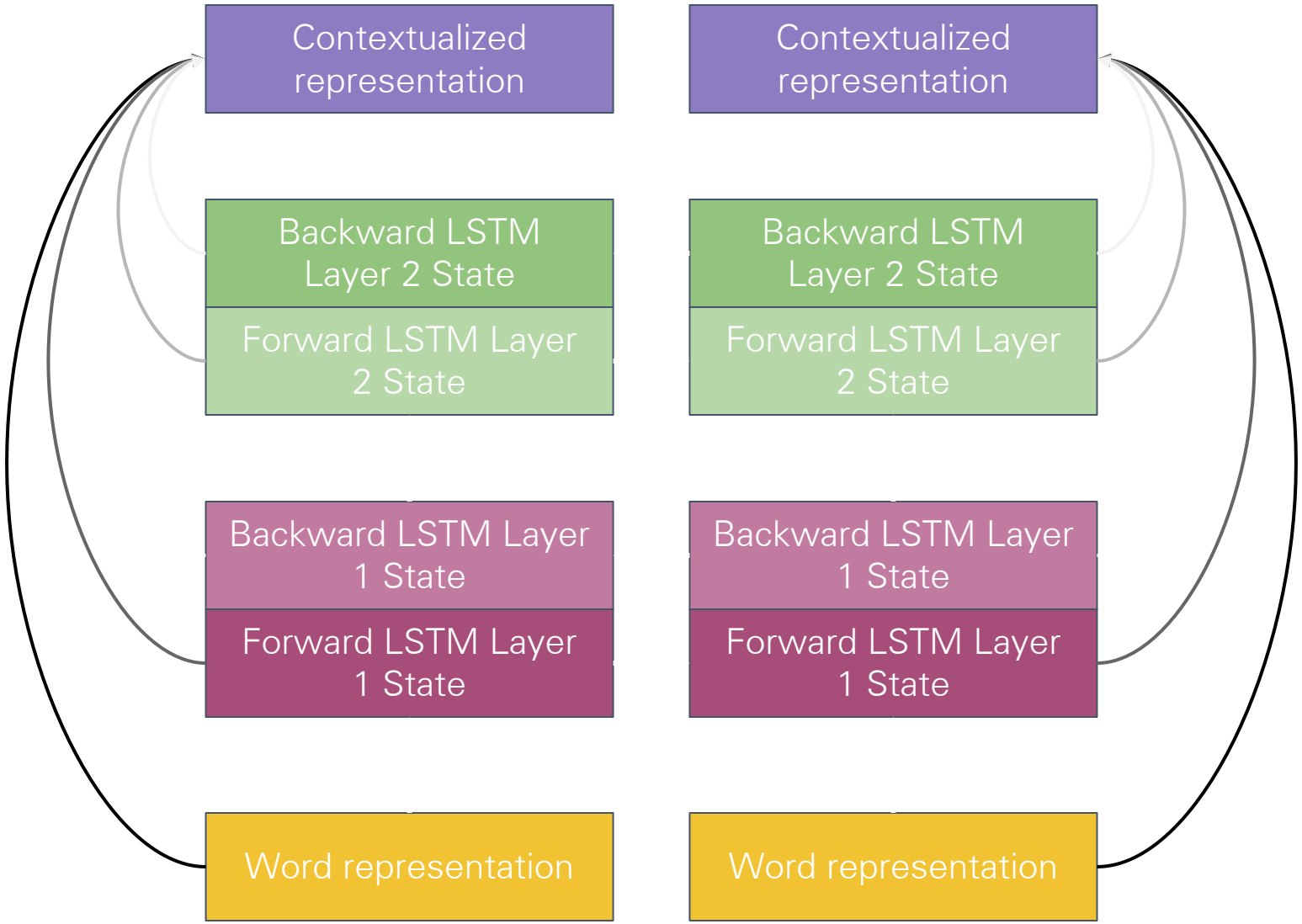
- Replace word vectors with a learned weighted sum of features of deep bi-directional LM
- Improves baseline models to SOTA
- Uses the LM from (Jozefowicz et al. 2016)
- Extends benefits of LMs to a much wider variety of tasks

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%



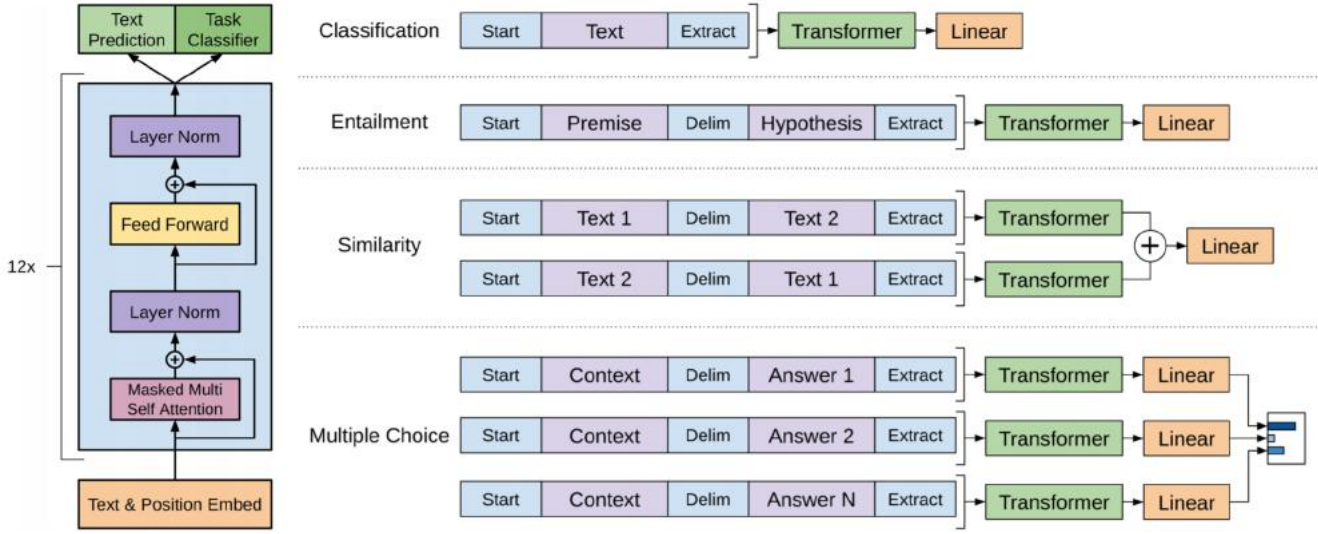
Deep contextualized word representations

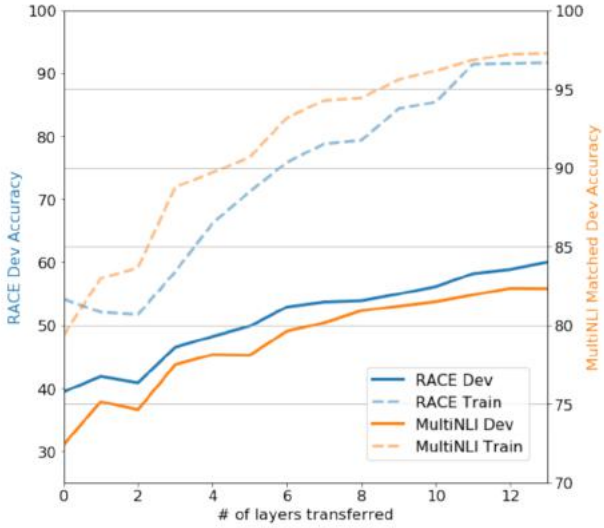
Peters et al. 2018

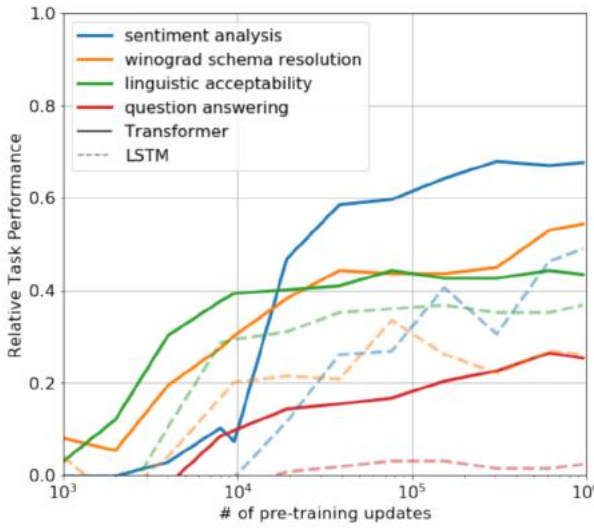


Improving Language Understanding by Generative Pre-Training (GPT-1)

- Transformer based LM
- 12 self-attention blocks - 12 heads - 768 dim state
  - ~100M params
- Trained on 7,000 books ~ 5 GB of text (BookCorpus Zhu et al 2015)
- Fine-tune on supervised tasks (like Dai et al. 2015)
- Removes the need for task specific architectures







Improving Language Understanding by Generative Pre-Training (GPT-1)

DATASET	TASK	SOTA	OURS
SNLI	Textual Entailment	89.3	**89.9**
MNLI Matched	Textual Entailment	80.6	**82.1**
MNLI Mismatched	Textual Entailment	80.1	**81.4**
SciTail	Textual Entailment	83.3	**88.3**
QNLI	Textual Entailment	82.3	**88.1**
RTE	Textual Entailment	**61.7**	56.0
STS-B	Semantic Similarity	81.0	**82.0**
QQP	Semantic Similarity	66.1	**70.3**
MRPC	Semantic Similarity	**86.0**	82.3
RACE	Reading Comprehension	53.3	**59.0**
ROCStories	Commonsense Reasoning	77.6	**86.5**
COPA	Commonsense Reasoning	71.2	**78.6**
SST-2	Sentiment Analysis	**93.2**	91.3
CoLA	Linguistic Acceptability	35.0	**45.4**
GLUE	Multi Task Benchmark	68.9	**72.8**

Lecture overview

- Motivation and Intro
- Introduction to Language Models
- History of Neural Language Models
- **A digression into Transformers**
- Beyond standard LMs
- Why we need Unsupervised Learning

Query what you want to look for

Key what you can compare to

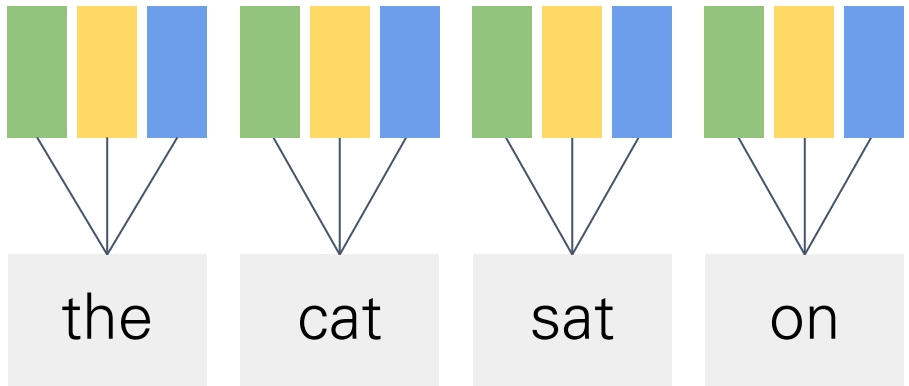
Value information you can retrieve

the	cat	sat	on

Query what you want to look for

Key what you can compare to

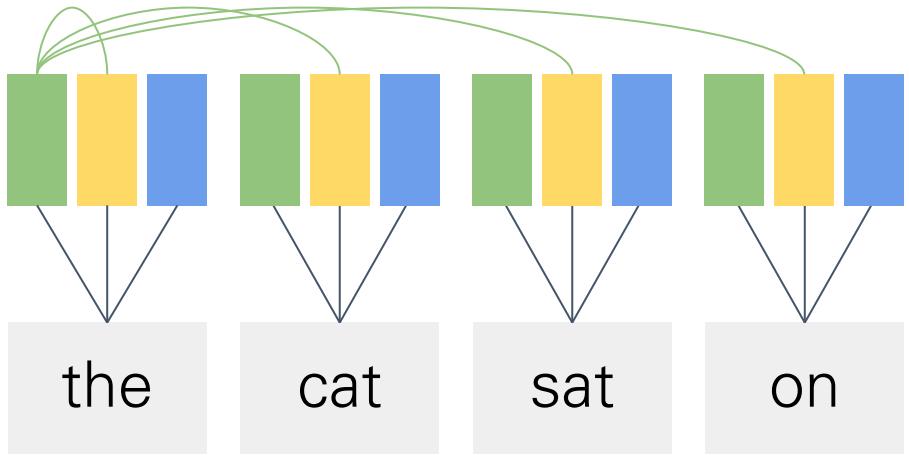
Value information you can retrieve



Query what you want to look for

Key what you can compare to

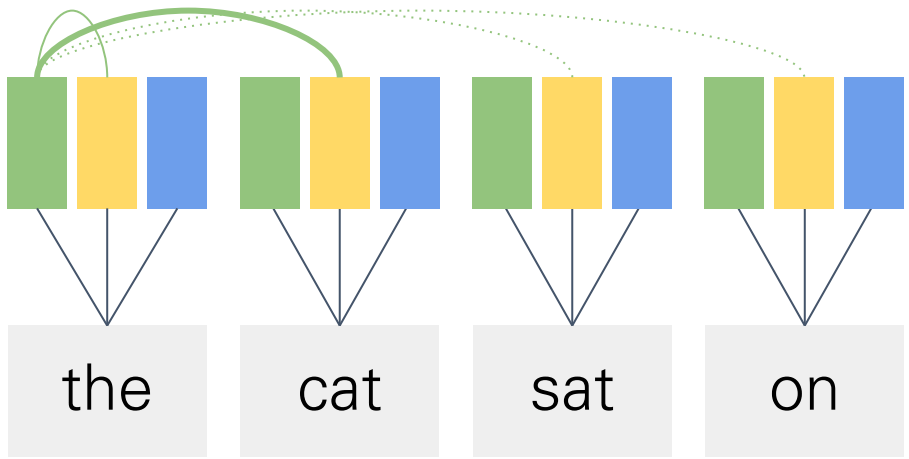
Value information you can retrieve



Query what you want to look for

Key what you can compare to

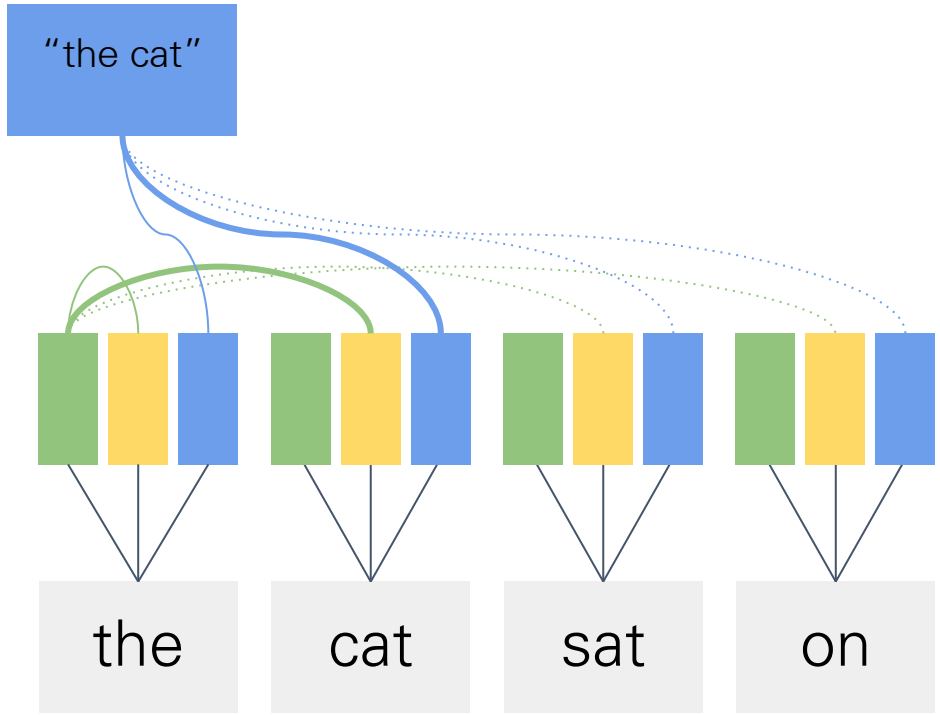
Value information you can retrieve

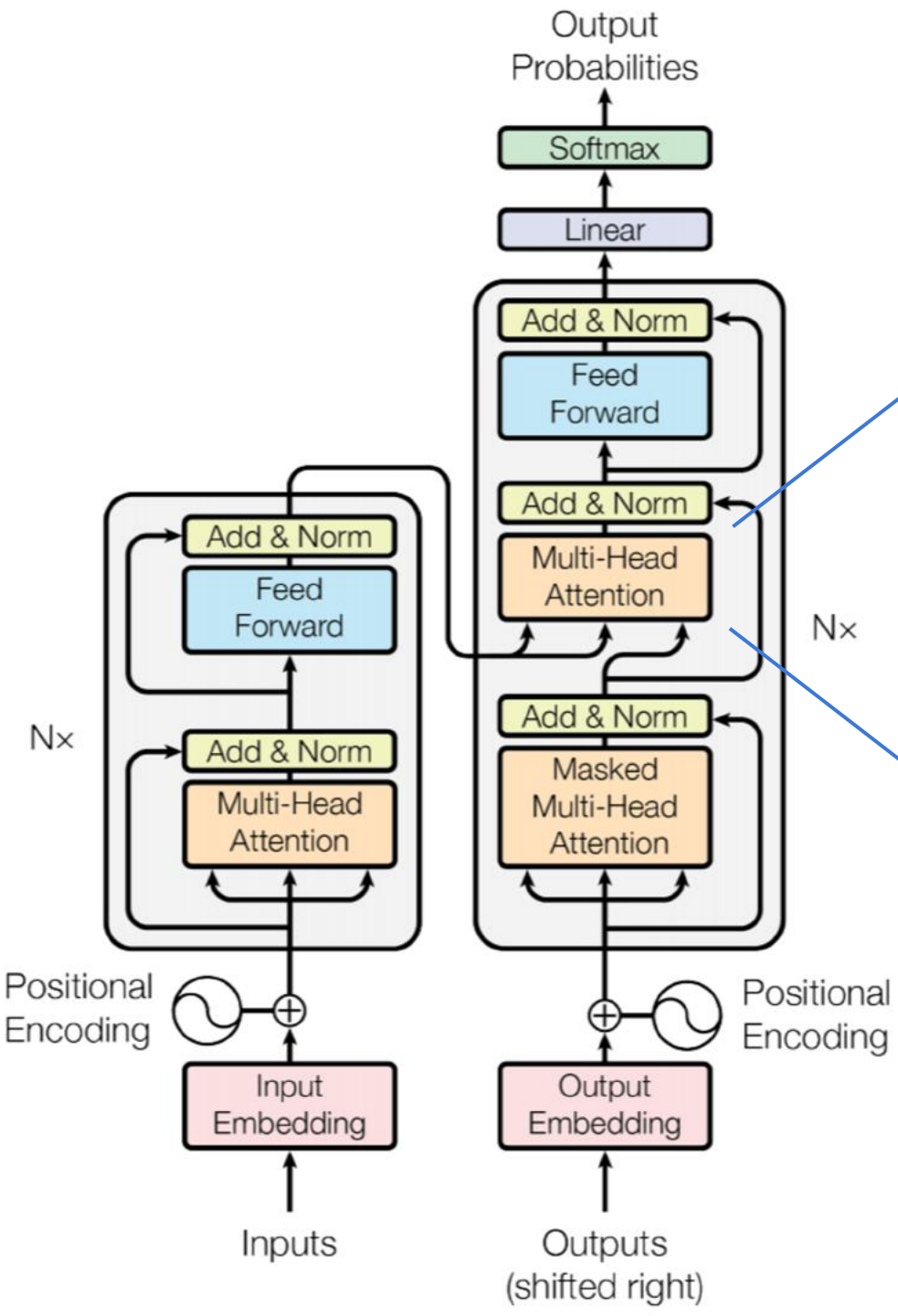


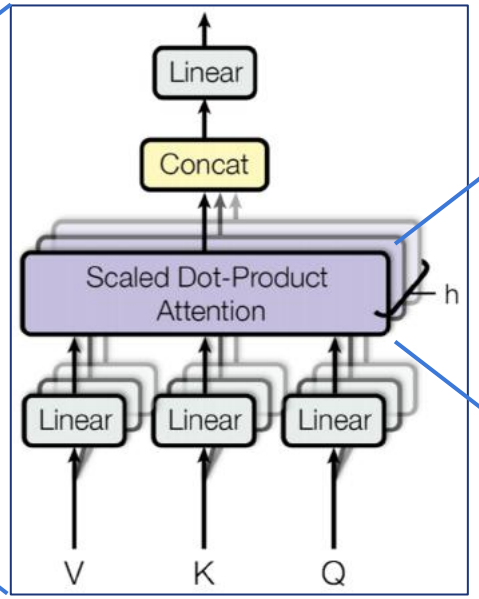
Query what you want to look for

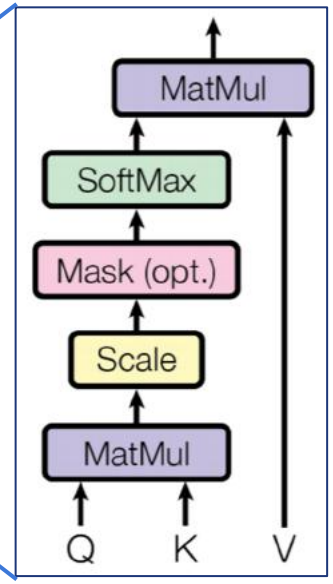
Key what you can compare to

Value information you can retrieve

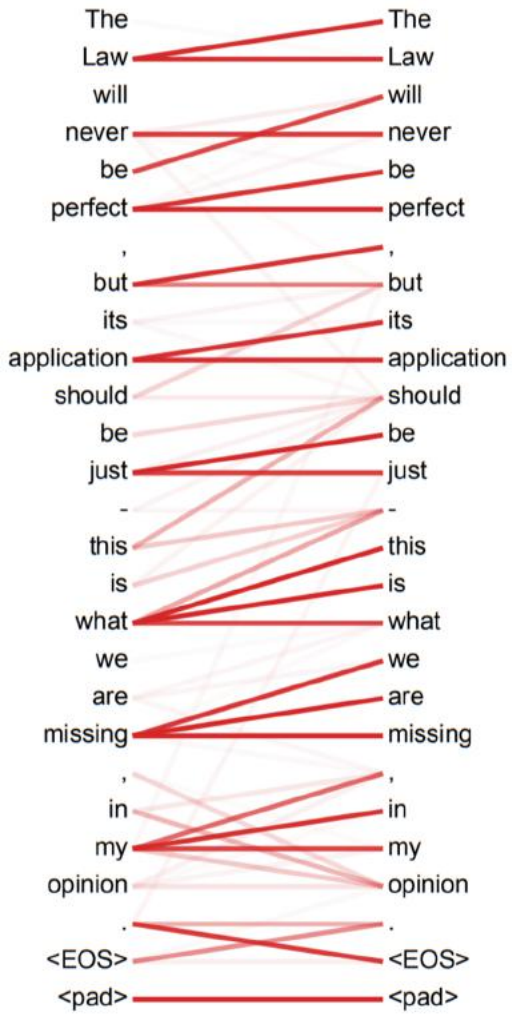


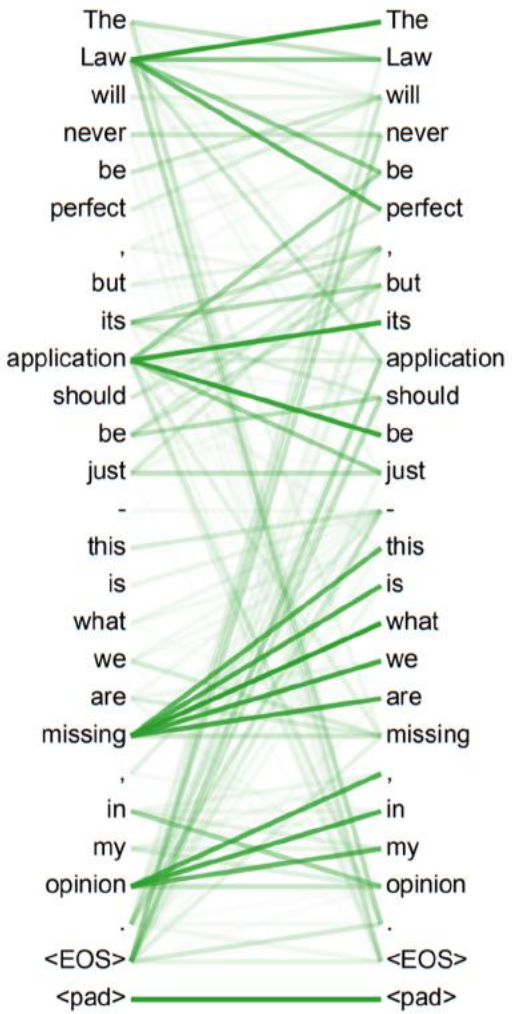


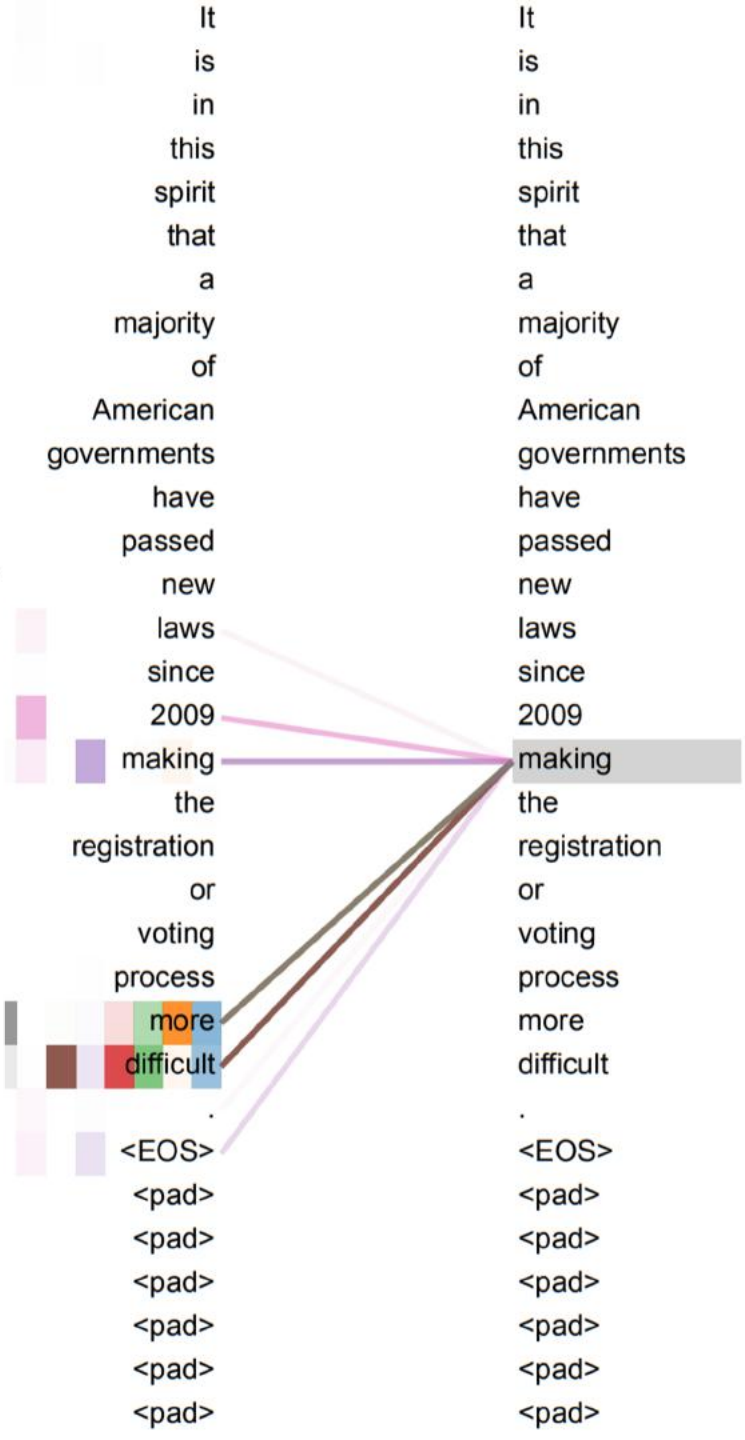


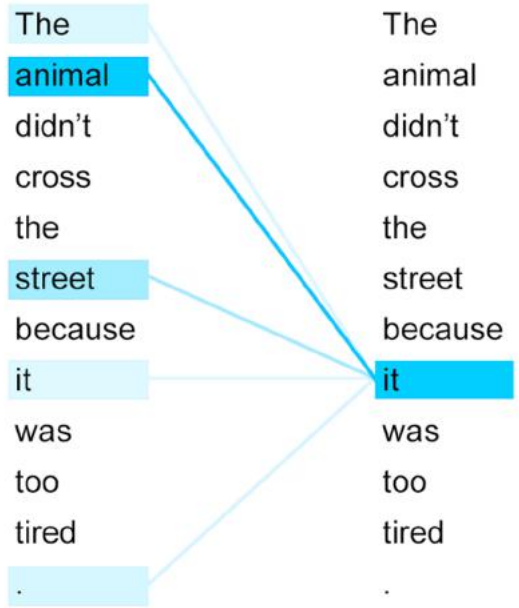


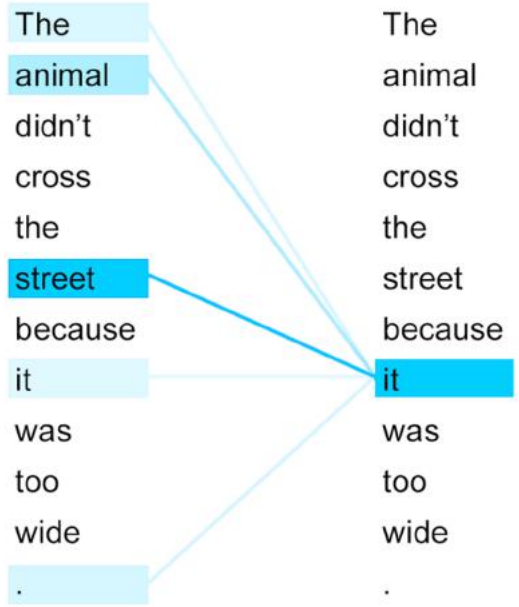
[Vaswani et al 2017]


The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>
The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>


The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>
The
Law
will
never
be
perfect
,
but
its
application
should
be
just
-
this
is
what
we
are
missing
,
in
my
opinion
.
<EOS>
<pad>


It
is
in
this
spirit
that
a
majority
of
American
governments
have
passed
new
laws
since
2009
making
the
registration
or
voting
process
more
difficult
.
<EOS>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
It
is
in
this
spirit
that
a
majority
of
American
governments
have
passed
new
laws
since
2009
making
the
registration
or
voting
process
more
difficult
.
<EOS>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>


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


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

Lecture overview

- Motivation and Intro
- Introduction to Language Models
- History of Neural Language Models
- A digression into Transformers
- **Beyond standard LMs**
- Why we need Unsupervised Learning

A lot of Improvements!

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
4	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	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
10	Facebook AI	RoBERTa		88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
12	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-
20	Jacob Devlin	BERT: 24-layers, 16-heads, 1024-hidden		80.5	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7	85.9	92.7	70.1	65.1	39.6
29	GLUE Baselines	BiLSTM+ELMo+Attn		70.0	33.6	90.4	84.4/78.0	74.2/72.3	63.1/84.3	74.1	74.5	79.8	58.9	65.1	21.7

MultiNLI

Premise: Hills and mountains are especially sanctified in Jainism.

Hypotheis: Janism hates nature.

Label: Contradiction

CoLa

Sentence: The wagon rumbled down the road.

Label: Acceptable

Sentence: The car honked down the road.

Label: Unacceptable

A lot of Improvements!

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
4	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	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
10	Facebook AI	RoBERTa		88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
12	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-
20	Jacob Devlin	BERT: 24-layers, 16-heads, 1024-hidden		80.5	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7	85.9	92.7	70.1	65.1	39.6
29	GLUE Baselines	BiLSTM+ELMo+Attn		70.0	33.6	90.4	84.4/78.0	74.2/72.3	63.1/84.3	74.1	74.5	79.8	58.9	65.1	21.7

More on this later!

MultiNLI

Premise: Hills and mountains are especially sanctified in Jainism.

Hypotheis: Janism hates nature.

Label: Contradiction

CoLa

Sentence: The wagon rumbled down the road.

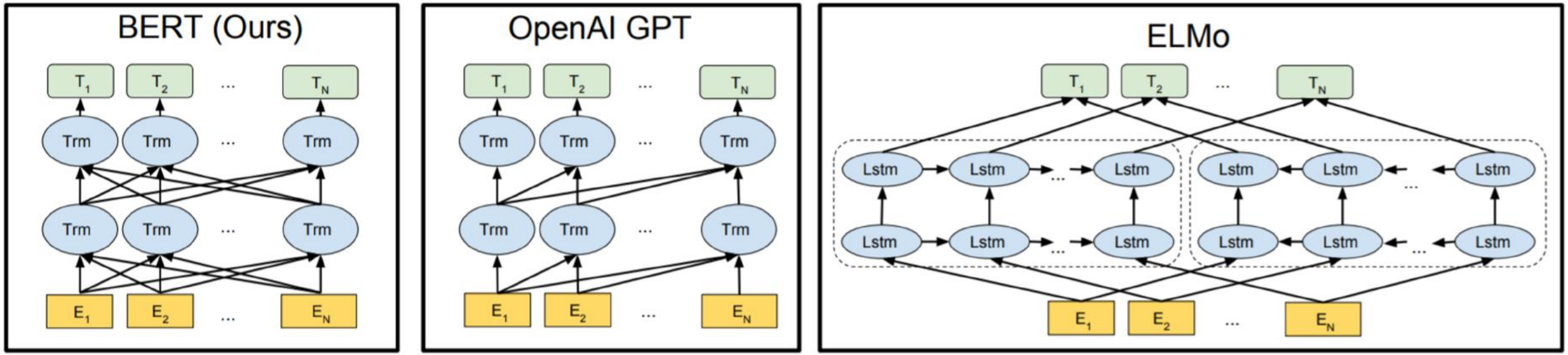
Label: Acceptable

Sentence: The car honked down the road.

Label: Unacceptable

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

Devlin et al. 2017

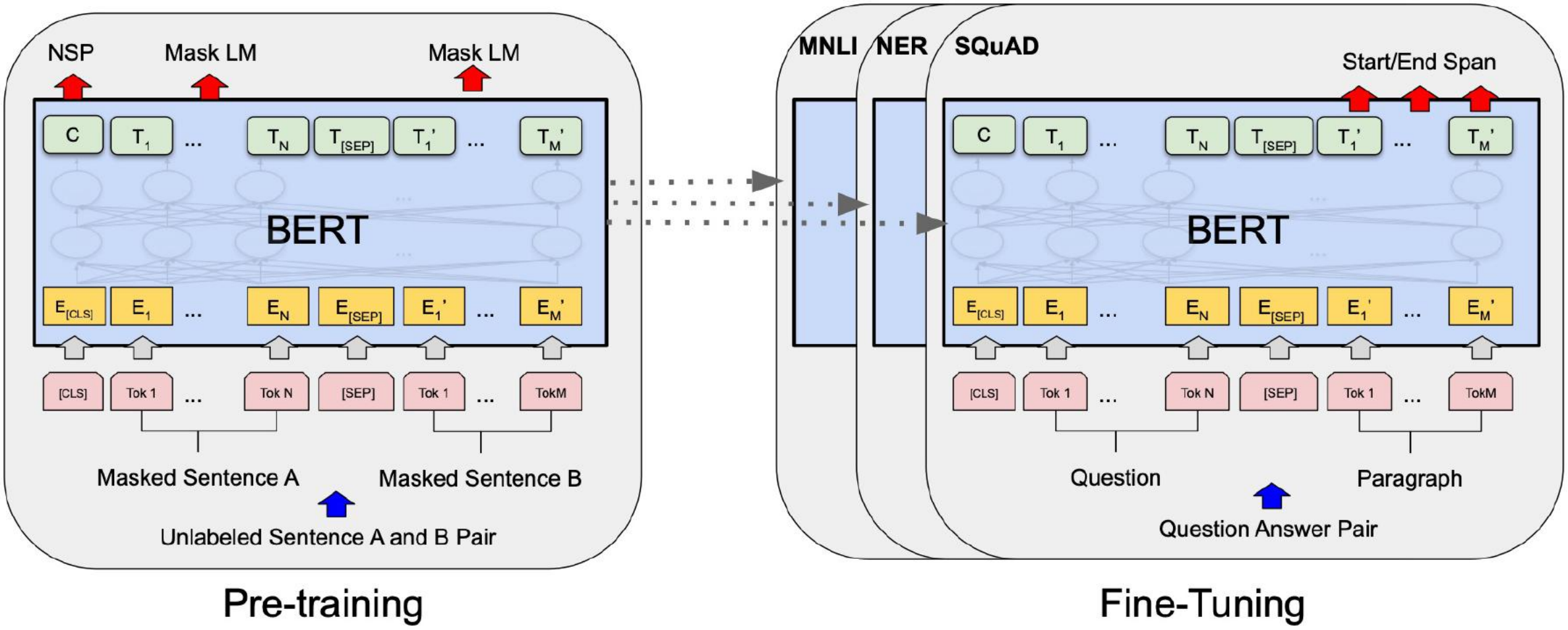



```
Left-Right LM: The cat sat on the [mask] -> The cat sat on the mat
Right-Left LM: [mask] cat sat on the mat -> The cat sat on the mat
Masked LM:  The [mask] sat on the [mask] -> The cat sat on the mat
```

BERT Workflow

- The BERT workflow includes:
  - Pretrain on generic, self-supervised tasks, using large amounts of data (like all of Wikipedia)
  - Fine-tune on specific tasks with limited, labelled data.
- The pretraining tasks (will talk about this in more detail later):
  - Masked Language Modelling (to learn contextualized token representations)
  - Next Sentence Prediction (summary vector for the whole input)

BERT Architecture



BERT Architecture

Properties:

- Two input sequences.
  - Many NLP tasks have two inputs (question answering, paraphrase detection, entailment detection etc.)
- Computes embeddings
  - Both token, position and segment embeddings.
  - Special start and separation tokens.
- Architecture
  - Basically the same as transformer encoder.
- Outputs:
  - Contextualized token representations.
  - Special tokens for context.

BERT Embeddings

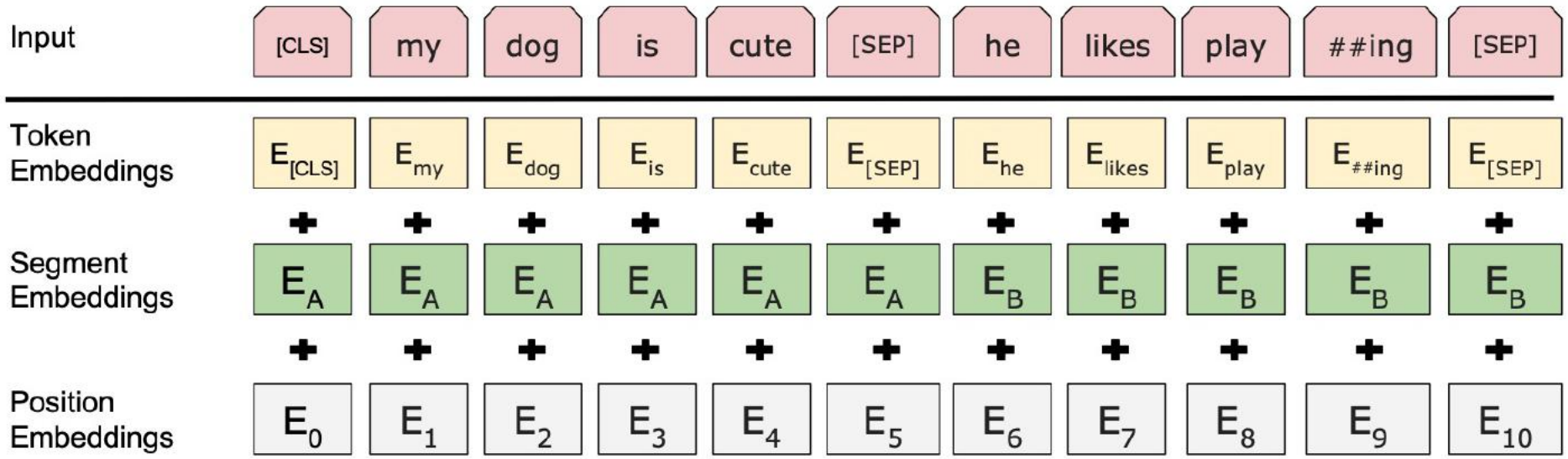


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

- How we tokenize the inputs is very important!
- BERT uses the WordPiece tokenizer (Wu et. al. 2016)

(Aside) Tokenizers

- Tokenizers have to balance the following:
  - Being comprehensive (rare words? translation to different languages)
  - Total number of tokens
  - How semantically meaningful each token is.
- This is an activate area of research.

Pretraining tasks

- Masked Language Modelling, i.e. Cloze Task (Taylor, 1953)
- Next sentence prediction

Masked Language Modelling

- Mask 15% of the input tokens. (i.e. replace with a dummy masking token)
- Run the model, obtain the embeddings for the masked tokens.
- Using these embeddings, try to predict the missing token.
- ”I love to eat peanut ___ and jam. ” Can you guess what’s missing?

- **This procedure forces the model to encode context information in the features of all of the tokens.**

Next Sentence Prediction

- Goal is to summarize the complete context (i.e. the two segments) in a single feature vector.
- Procedure for generating data
  - Pick a sentence from the training corpus and feed it as "segment A".
  - With 50% probability, pick the following sentence and feed that as "segment B".
  - With 50% probability, pick the a random sentence and feed it as "segment B".
- Using the features for the context token, predict whether segment B is the following sentence of segment A.
- Turns out to be a very effective pretraining technique!

Fine Tuning

Procedure:

- Add a final layer on top of BERT representations.
- Train the whole network on the fine-tuning dataset.
- Pre-training time: In the order of days on TPUs.
- Fine tuning task: Takes only a few hours max.

Fine Tuning

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

RoBERTa: A Robustly Optimized BERT Pretraining Approach

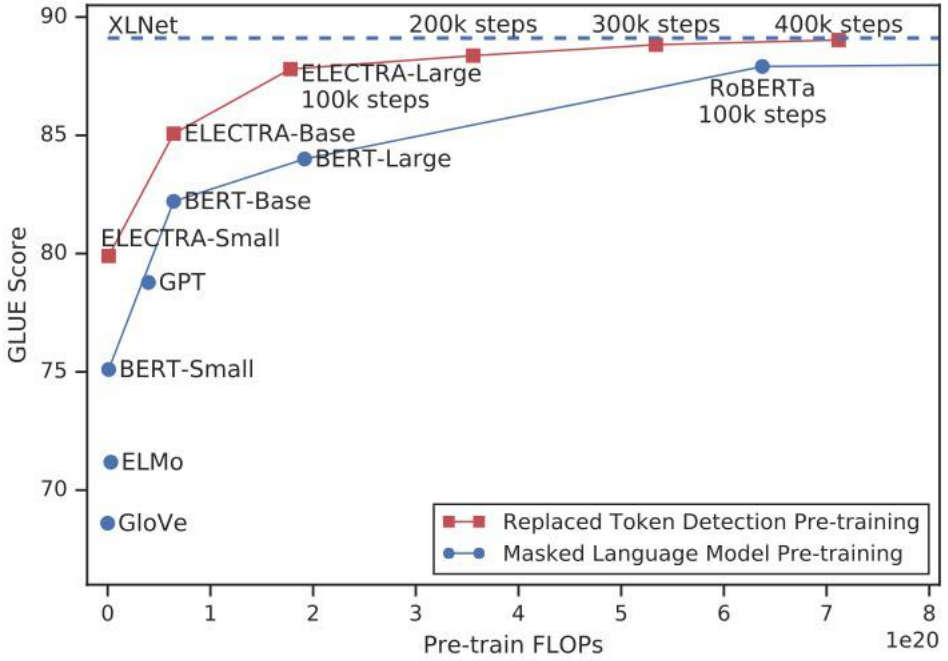
Liu et al. 2019

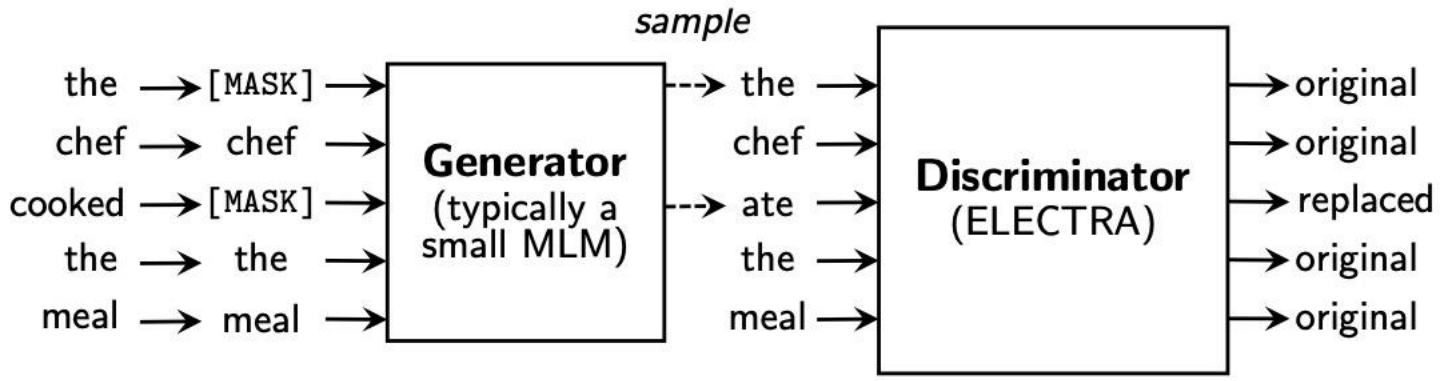
Really well executed refinement / engineering on BERT

- Better tuned (many HPs)
- Remove a few hacks (remove annealing context size)
- Better data generation (online instead of cached)
- A more flexible vocab scheme (more on this later)
- Use more compute / train longer (but same model capacity – BERT was undertrained)

ELECTRA

Clark et al. 2017





Model	Train / Infer FLOPs	Speedup	Params	Train Time + Hardware	GLUE
ELMo	3.3e18 / 2.6e10	19x / 1.2x	96M	14d on 3 GTX 1080 GPUs	71.2
GPT	4.0e19 / 3.0e10	1.6x / 0.97x	117M	25d on 8 P6000 GPUs	78.8
BERT-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	75.1
BERT-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	82.2
ELECTRA-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	79.9
50% trained	7.1e17 / 3.7e9	90x / 8x	14M	2d on 1 V100 GPU	79.0
25% trained	3.6e17 / 3.7e9	181x / 8x	14M	1d on 1 V100 GPU	77.7
12.5% trained	1.8e17 / 3.7e9	361x / 8x	14M	12h on 1 V100 GPU	76.0
6.25% trained	8.9e16 / 3.7e9	722x / 8x	14M	6h on 1 V100 GPU	74.1
ELECTRA-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	85.1

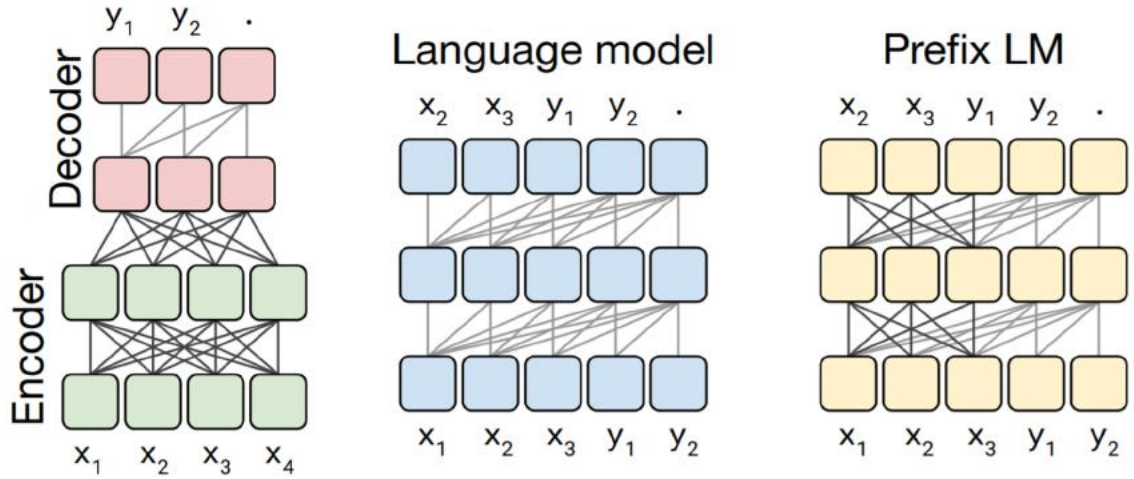
T5: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

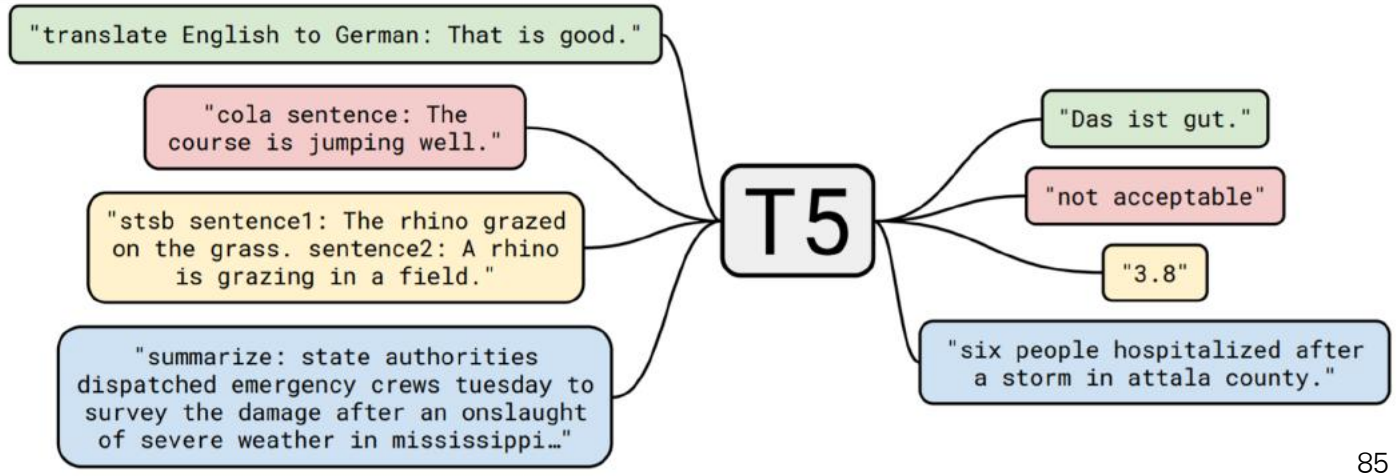
Raffel et al. 2019

- Very thorough (50 pages!) exploration of the design space of pretraining with a pleasing task formulation (from McCann et al 2018)

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style	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)*
I.i.d. noise, mask tokens	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>

Architecture	Objective	Params	Cost	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Encoder-decoder	Denoising	$2P$	M	**83.28**	**19.24**	**80.88**	**71.36**	**26.98**	**39.82**	**27.65**
Enc-dec, shared	Denoising	P	M	82.81	18.78	**80.63**	**70.73**	26.72	39.03	**27.46**
Enc-dec, 6 layers	Denoising	P	$M/2$	80.88	18.97	77.59	68.42	26.38	38.40	26.95
Language model	Denoising	P	M	74.70	17.93	61.14	55.02	25.09	35.28	25.86
Prefix LM	Denoising	P	M	81.82	18.61	78.94	68.11	26.43	37.98	27.39
Encoder-decoder	LM	$2P$	M	79.56	18.59	76.02	64.29	26.27	39.17	26.86
Enc-dec, shared	LM	P	M	79.60	18.13	76.35	63.50	26.62	39.17	27.05
Enc-dec, 6 layers	LM	P	$M/2$	78.67	18.26	75.32	64.06	26.13	38.42	26.89
Language model	LM	P	M	73.78	17.54	53.81	56.51	25.23	34.31	25.38
Prefix LM	LM	P	M	79.68	17.84	76.87	64.86	26.28	37.51	26.76





Lecture overview

- Motivation and Intro
- Introduction to Language Models
- History of Neural Language Models
- A digression into Transformers
- Beyond standard LMs
- **Why we need Unsupervised Learning**

How well does supervised learning work?

- Natural Language Inference - SNLI (Bowman et al. 2015)
  - Predict logical relation between two sentences - P and H.
  - Contradiction -> A man inspects a uniform. A man is sleeping.
  - Neutral -> An older and younger man smiling. Two men are smiling at cats playing on the floor.
  - Entailment -> A soccer game with multiple males playing. Some men are playing a sport.
- Models are near human level according to the standard test set
- Humans ~ 88.0%
- ESIM (Chen et al. 2017) ~ 88.0%

Annotation Artifacts In Natural Language Inference Data

Gururangan et al. 2018

- Turkers were paid to create the training data of SNLI
  - They often use a few tricks or heuristics to quickly make data
- For instance:
  - Words like (not, never, nothing) hint at negation
  - Generic words like (person, animal, sport) hint at entailment
  - Modifiers like (tall, sad, popular) hint at neutral
- If you train a classifier on only the second sentence...
  - You get ~67.0% compared to ~33.0%
- ESIM performance drops from ~88% to ~72% on the hard examples

Breaking NLI Systems with Sentences that Require Simple Lexical Inferences

Glockner et al. 2018

Use known relations between words to construct a new test set

`The man is holding a {object}.`
`The man is holding a {different object}.`
Contradiction

`A little girl is very {adjective}.`
`A little girl is very {synonym}.`
Entailment

Built a new test set of 8,000 examples from 14 categories to probe this.

ESIM drops from ~88% to ~66% on this new test set

Learning and Evaluating General Linguistic Intelligence

Yogatama et al. 2019

- Near SOTA QA model (BERT on SQUAD) drops from 86.5 F1 to:

- 35.6 F1 on TriviaQA
- 56.2 F1 on QuAC

What might be going wrong?

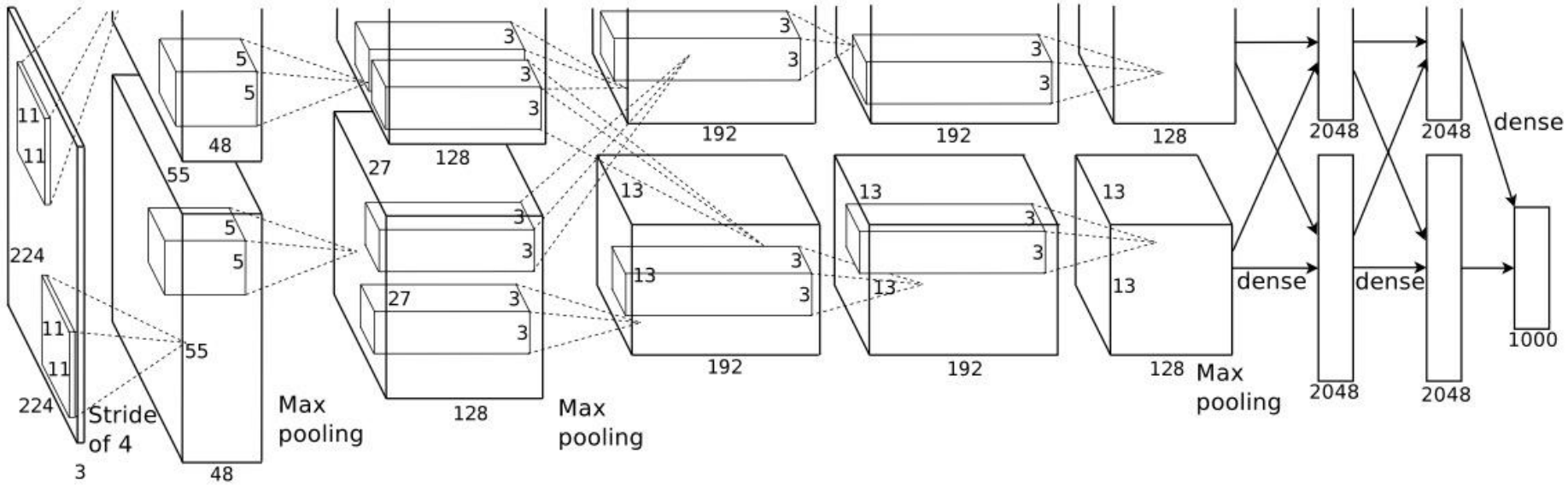
- Standard training datasets might not encourage generalization
- Models learn spurious associations in the training set
- Models exploit distributional bias of the creation of the training set
- Models “stop learning” once they get to 0 training error
- Current techniques are brittle
- Current techniques are closer to memorization than generalization

How to make progress?

- Better models / architectures?
- More data?
- Different paths all together?

How to make progress?

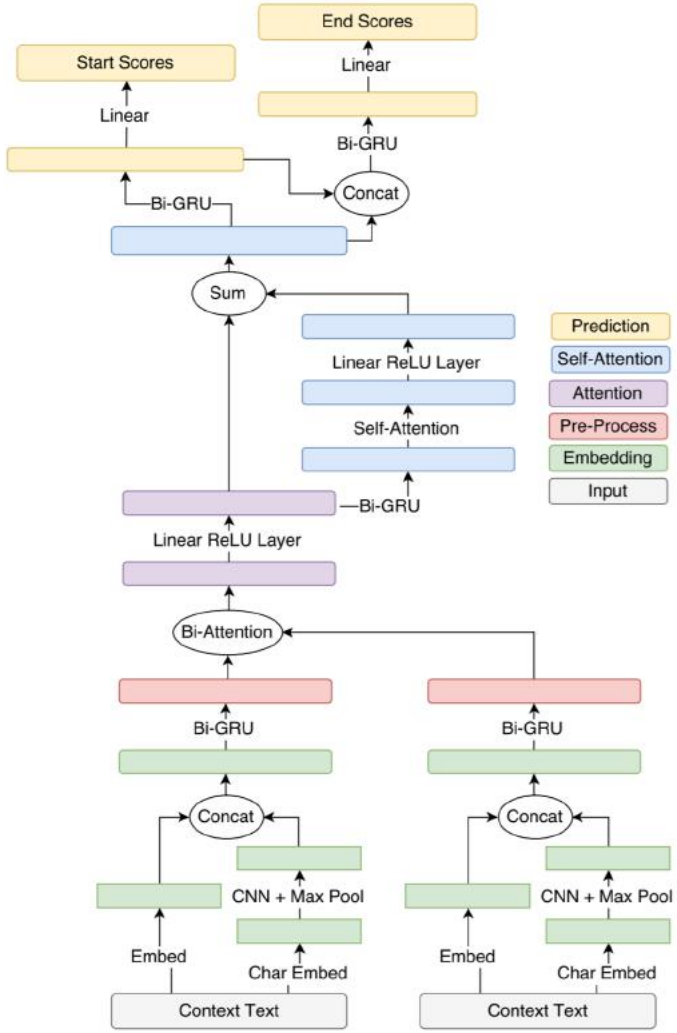
The beautiful story of modern deep learning was supposed to be that we cleverly encoded high-level domain knowledge into our architectures and built these larger labeled datasets and then let SGD figure out all the annoying details for us.



How to make progress?

- This set us up for a mindset of architecture engineering.
- There's a very large design space:
  - Multiply by a sigmoid here
  - Add a temporal max-pool there
  - Convolve with not 1 (or 2) but three different width filters
  - Throw some attention on top of it all for good measure

We really like playing with blocks!

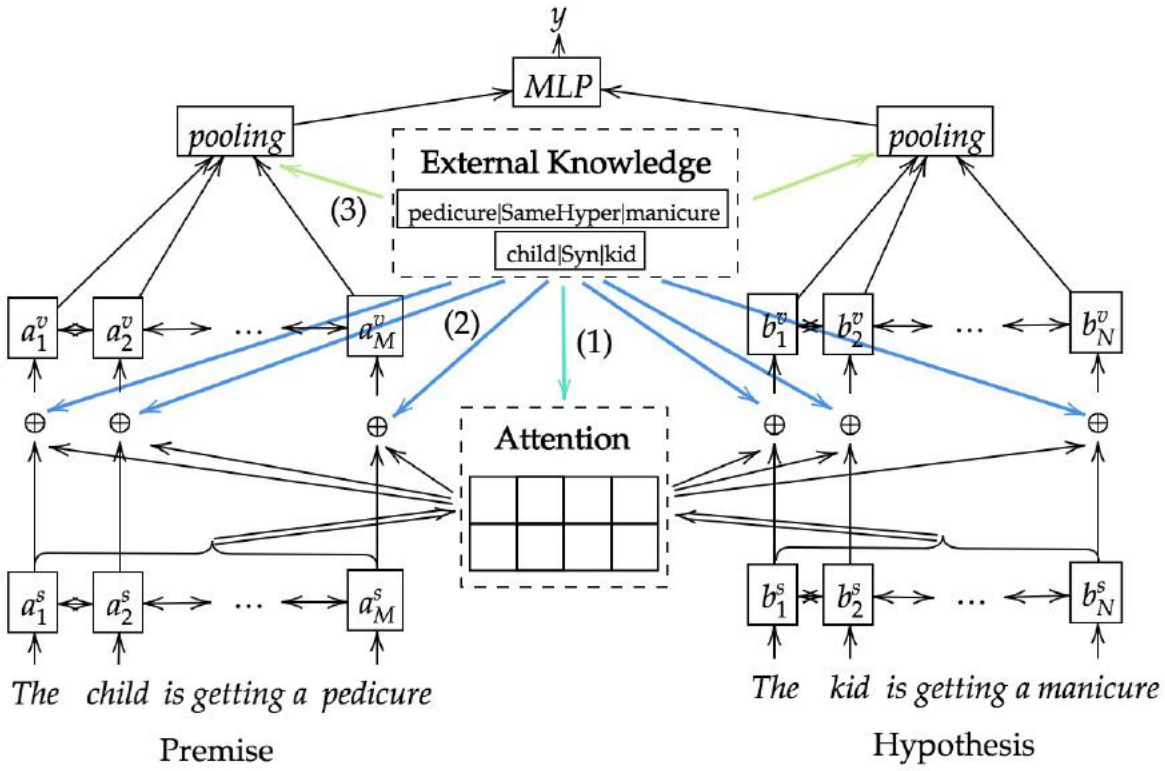


4. Knowledge Enriched Inference Composition

3. Knowledge Enriched Local Inference Collection

2. Knowledge Enriched Co – attention

1. Input Encoding

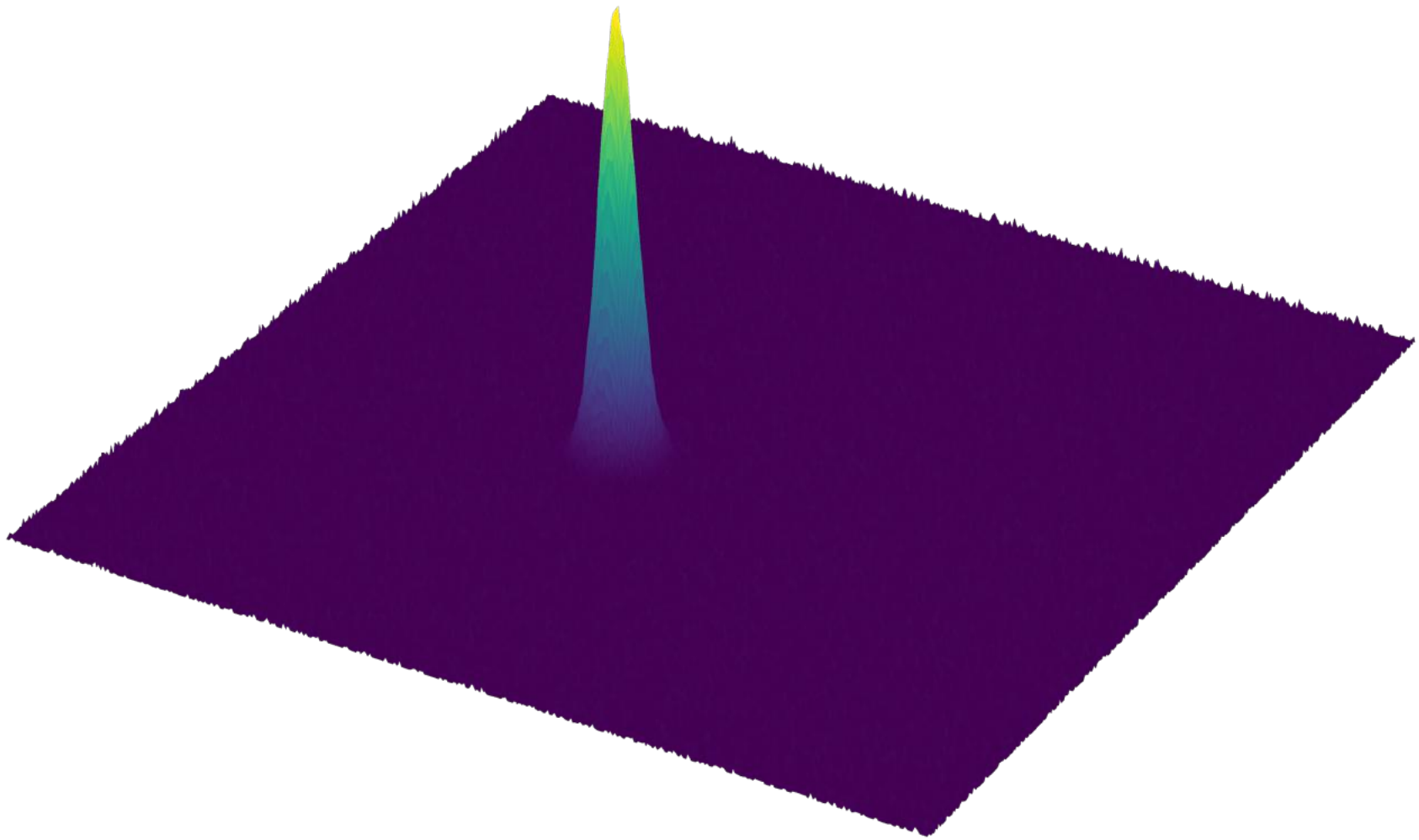


How to make progress?

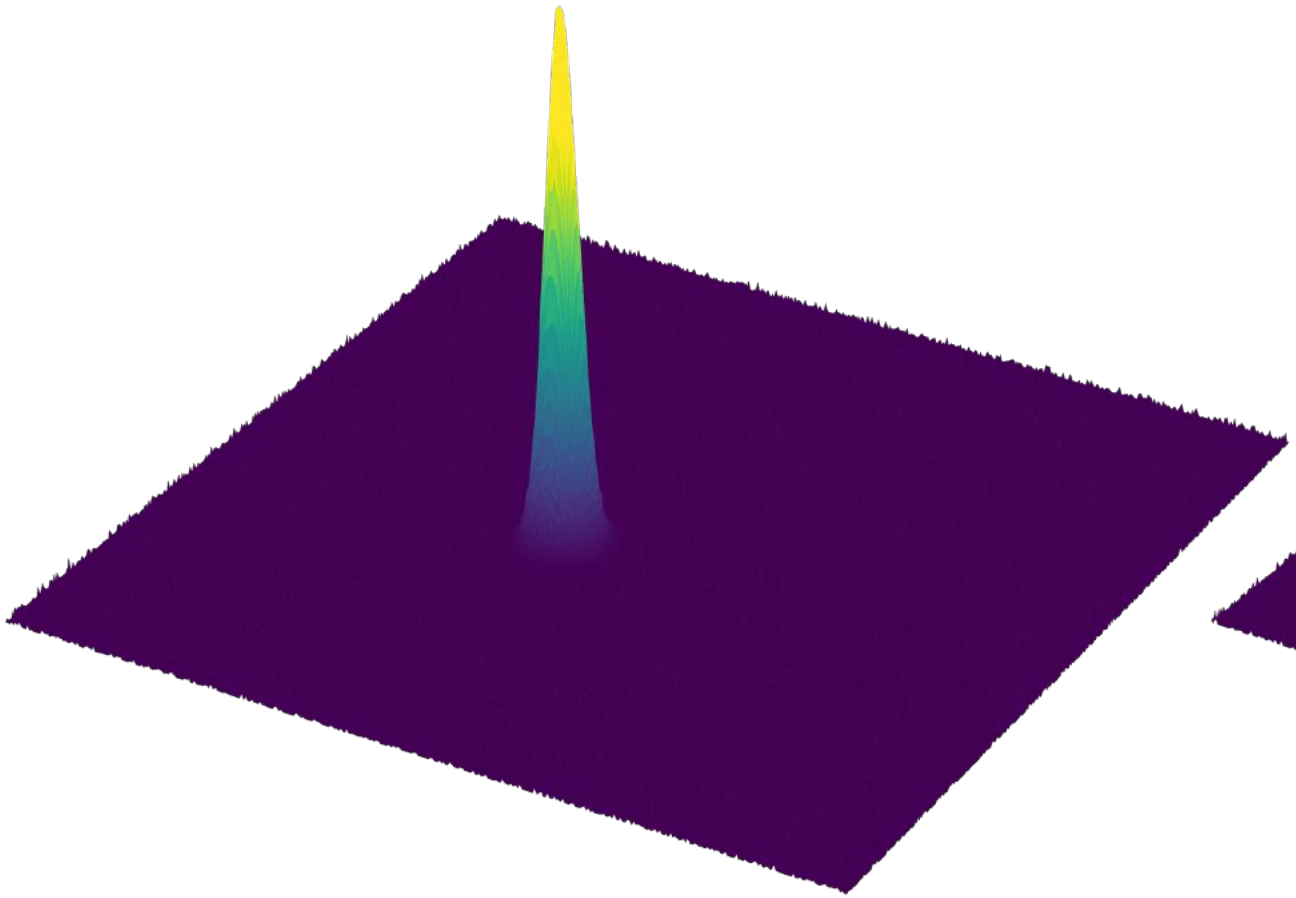
- We can encode useful information through the choice of model:
  - Convolution
  - Recurrence
  - Weight Sharing
  - Attention
  - Hierarchy
  - Depth

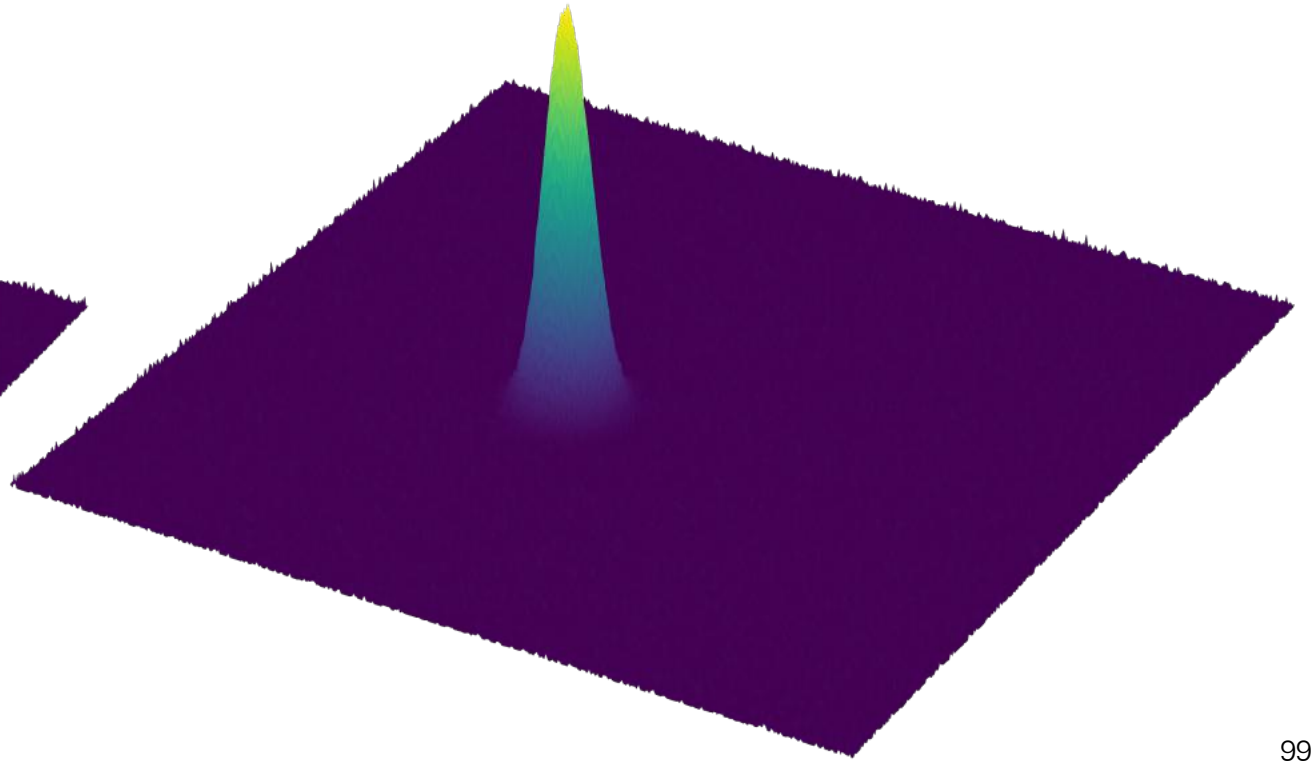
These are all important and impactful

What’s going on?



What we've been mostly doing





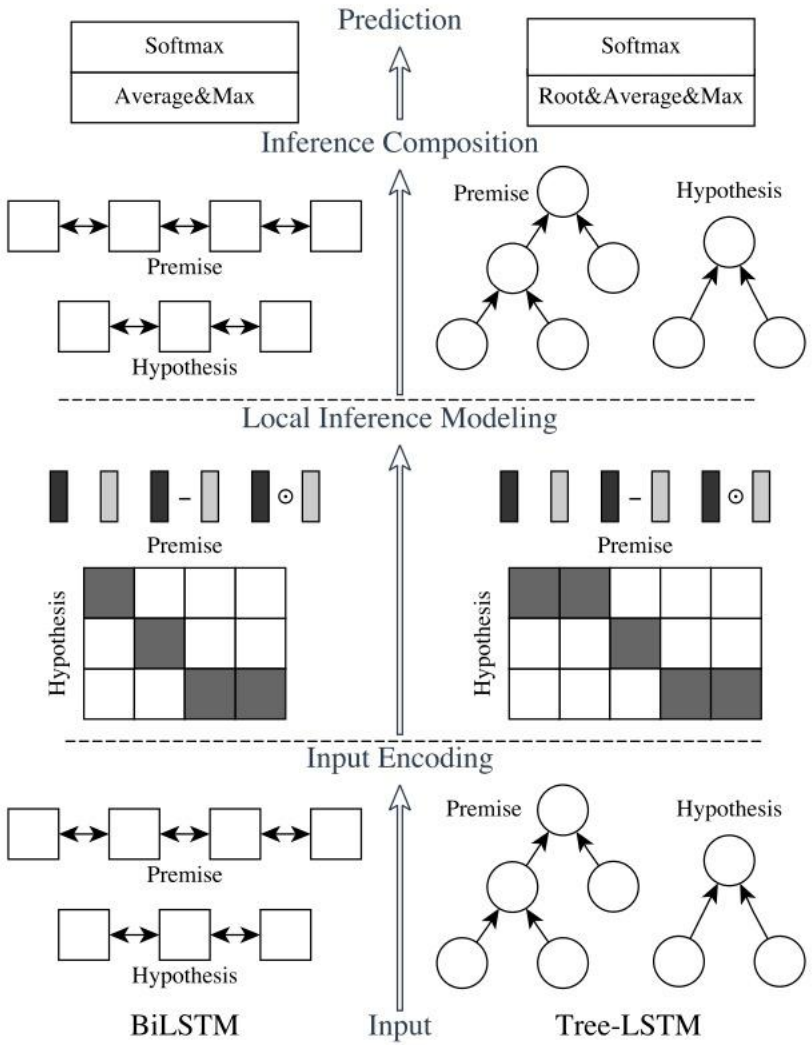
On the State of the Art of Evaluation in Neural Language Models

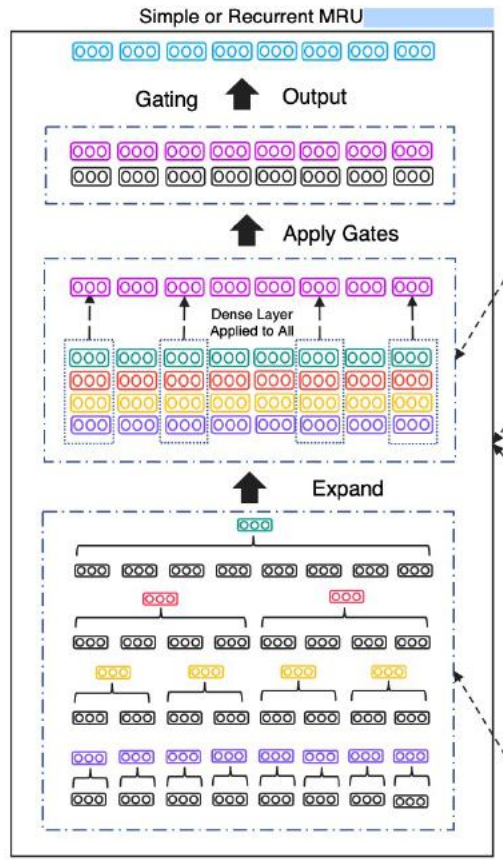
Gábor Melis[†], **Chris Dyer**[†], **Phil Blunsom**[†‡]
{melisgl,cdyer,pblunsom}@google.com
[†]DeepMind
[‡]University of Oxford

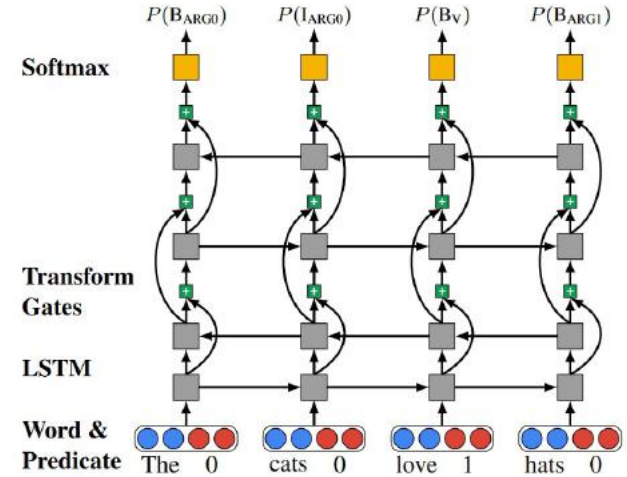
Abstract

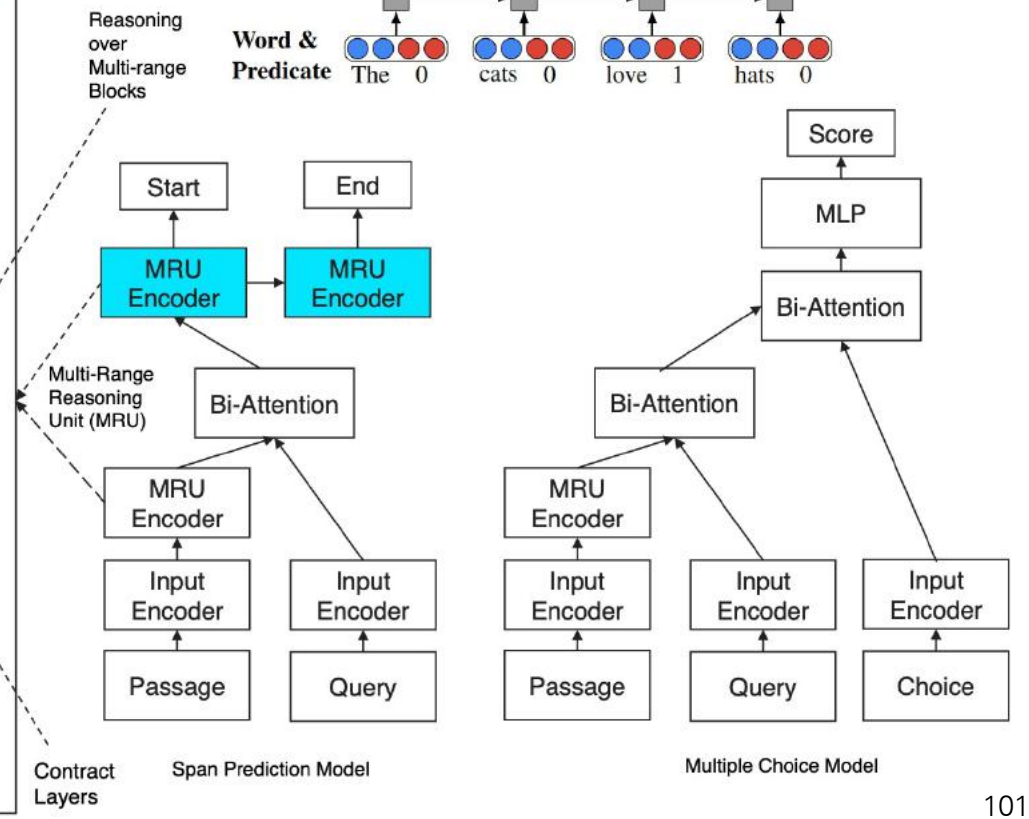
Ongoing innovations in recurrent neural network architectures have provided a steady influx of apparently state-of-the-art results on language modelling benchmarks. However, these have been evaluated using differing codebases and limited computational resources, which represent uncontrolled sources of experimental variation. We reevaluate several popular architectures and regularisation methods with large-scale automatic black-box hyperparameter tuning and arrive at the somewhat surprising conclusion that standard LSTM architectures, when properly regularised, outperform more recent models. We establish a new state of the art on the Penn Treebank and Wikitext-2 corpora, as well as strong baselines on the Hutter Prize dataset.

The value of architecture engineering?









How to learn?

- Supervised Learning is the dominant approach
- The largest supervised dataset is JFT-300M (Sun et al. 2017)
  - 300 million images
  - 18,000 classes

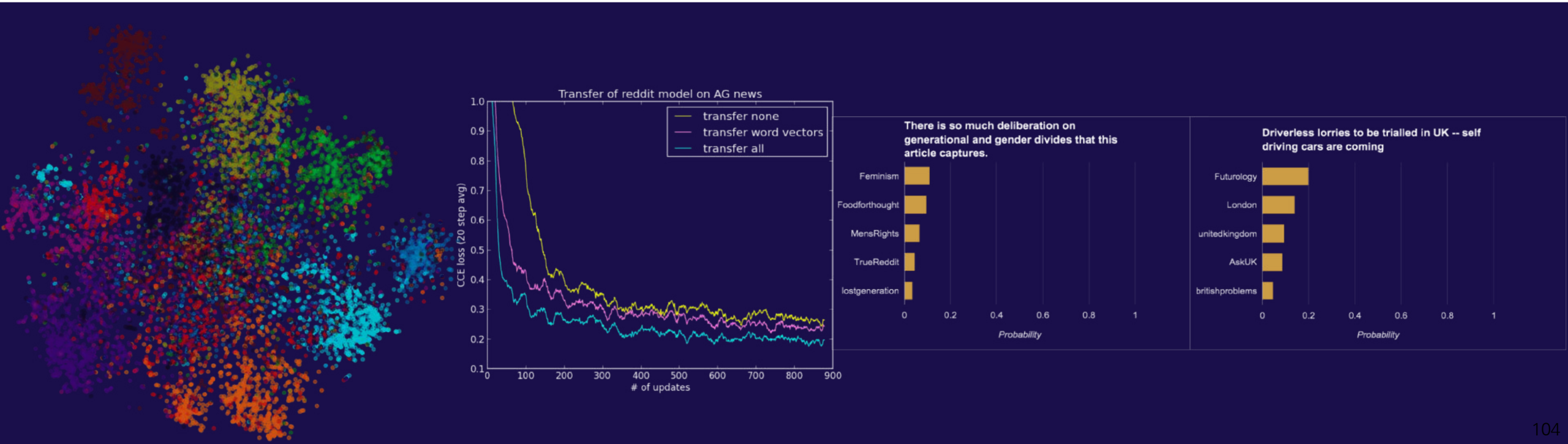
How to learn?

- Supervised Learning is the dominant approach
- The largest supervised dataset is JFT-300M (Sun et al. 2017)
  - 300 million images
  - 18,000 classes
- JFT is only 530 MB of constraints

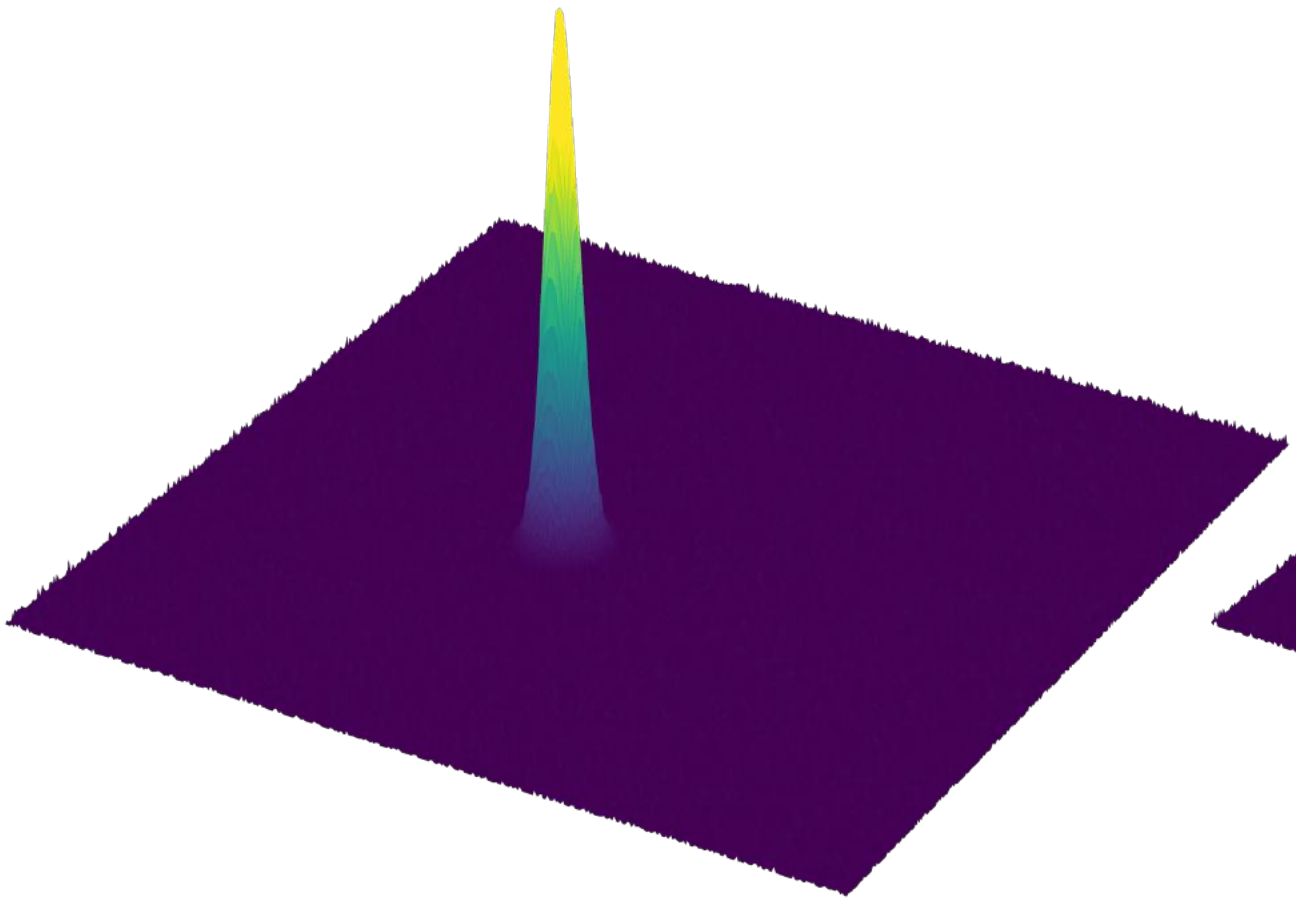
Pursuing this route for language

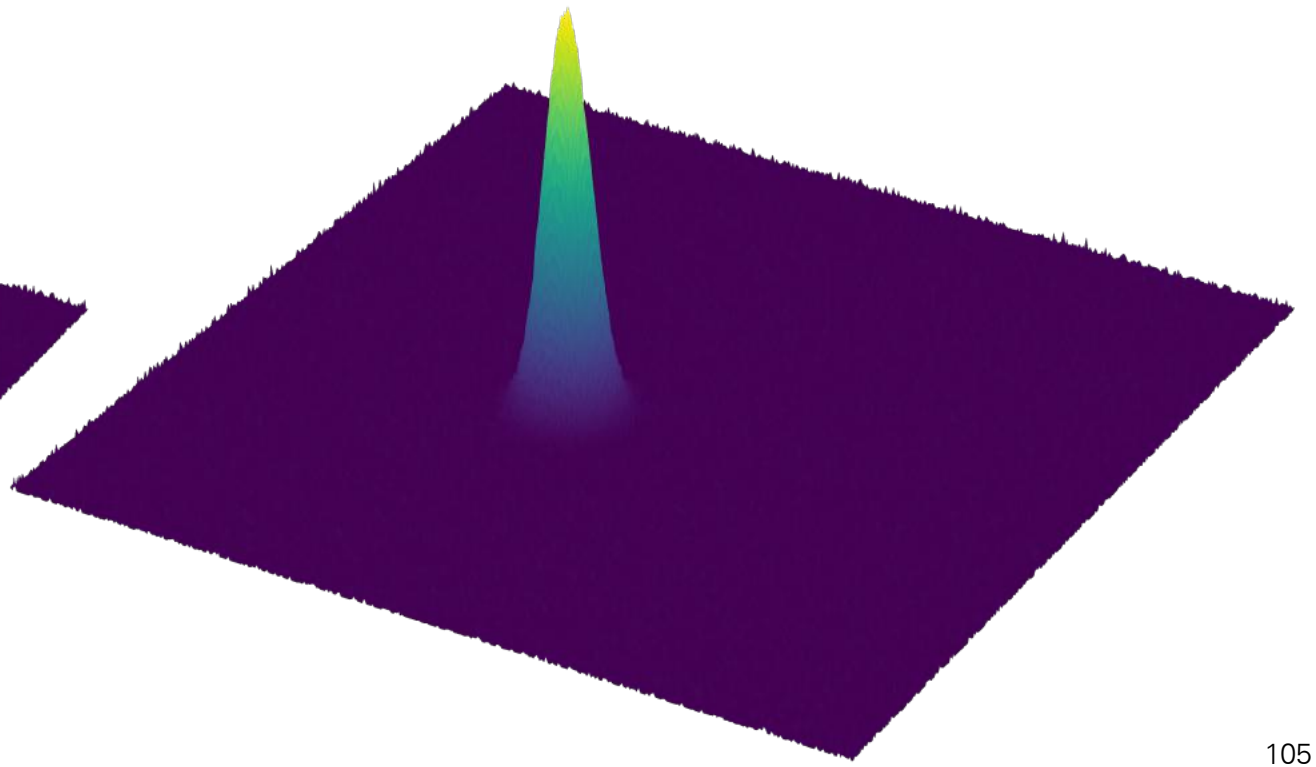
Spent most of 2015 trying to build what I hoped would be "Imagenet for text" (to enable impactful transfer learning)

- 20 Newsgroups but for reddit = giant weakly supervised dataset
  - 150M labeled examples across 1,000 communities
  - Trained RNNs to predict the community from the discussion

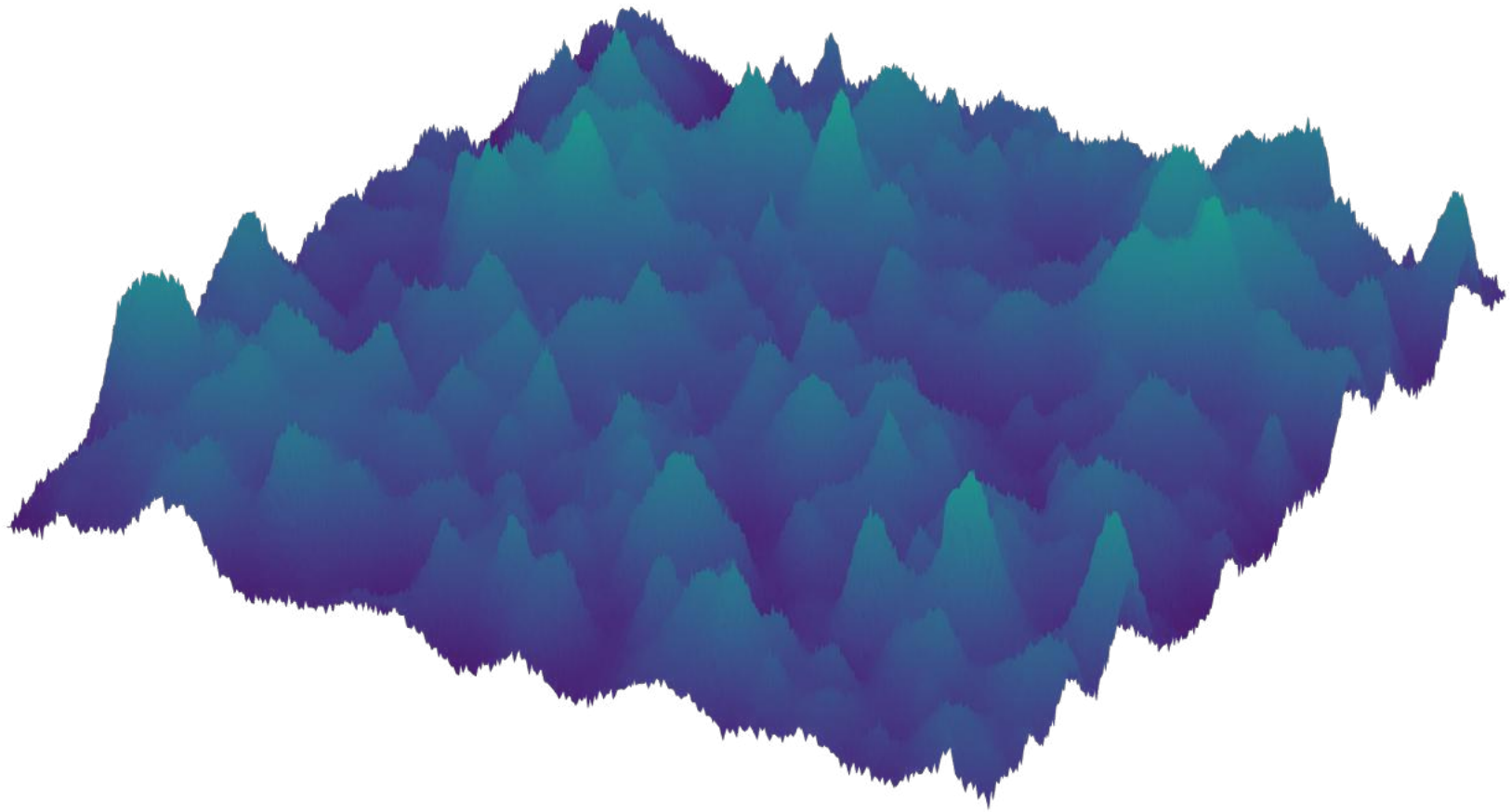


What we've been trying





How to do this instead?



Information and Representation Engineering alongside Architecture Engineering

- KIM (Chen et al. 2017)
- Gets 83.5% on the new NLI test set

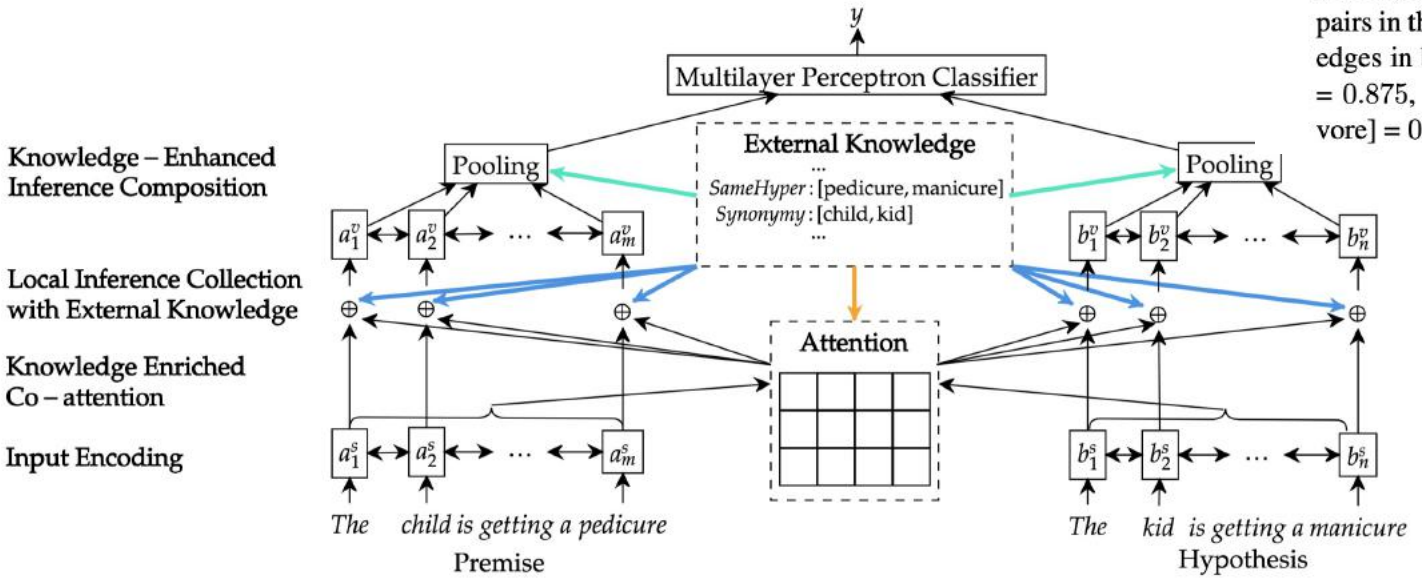
(1) *Synonymy*: It takes the value 1 if the words in the pair are synonyms in WordNet (i.e., belong to the same synset), and 0 otherwise. For example, [felicitous, good] = 1, [dog, wolf] = 0.

(2) *Antonymy*: It takes the value 1 if the words in the pair are antonyms in WordNet, and 0 otherwise. For example, [wet, dry] = 1.

(3) *Hypernymy*: It takes the value $1 - n/8$ if one word is a (direct or indirect) hypernym of the other word in WordNet, where n is the number of edges between the two words in hierarchies, and 0 otherwise. Note that we ignore pairs in the hierarchy which have more than 8 edges in between. For example, [dog, canid] = 0.875, [wolf, canid] = 0.875, [dog, carnivore] = 0.75, [canid, dog] = 0

(4) *Hyponymy*: It is simply the inverse of the hypernymy feature. For example, [canid, dog] = 0.875, [dog, canid] = 0.

(5) *Co-hyponyms*: It takes the value 1 if the two words have the same hypernym but they do not belong to the same synset, and 0 otherwise. For example, [dog, wolf] = 1.

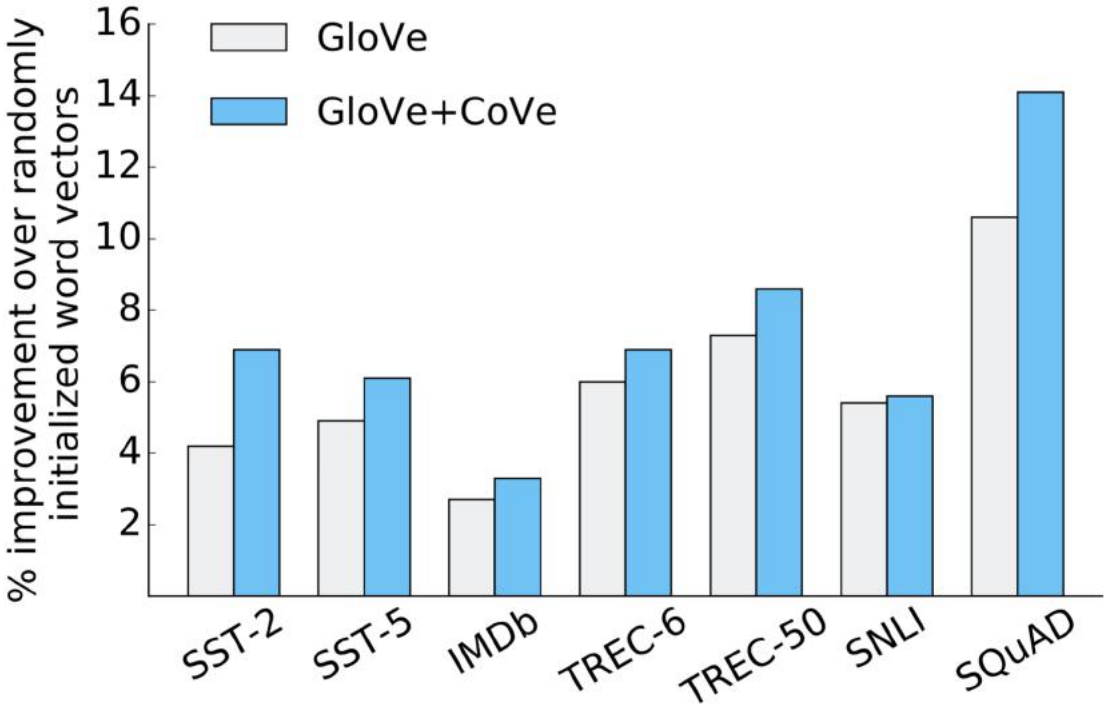


Feature	#Words	#Pairs
Synonymy	84,487	237,937
Antonymy	6,161	6,617
Hypernymy	57,475	753,086
Hyponymy	57,475	753,086
Co-hyponyms	53,281	3,674,700

Table 1: Statistics of lexical relation features.

Information and Representation Engineering alongside Architecture Engineering

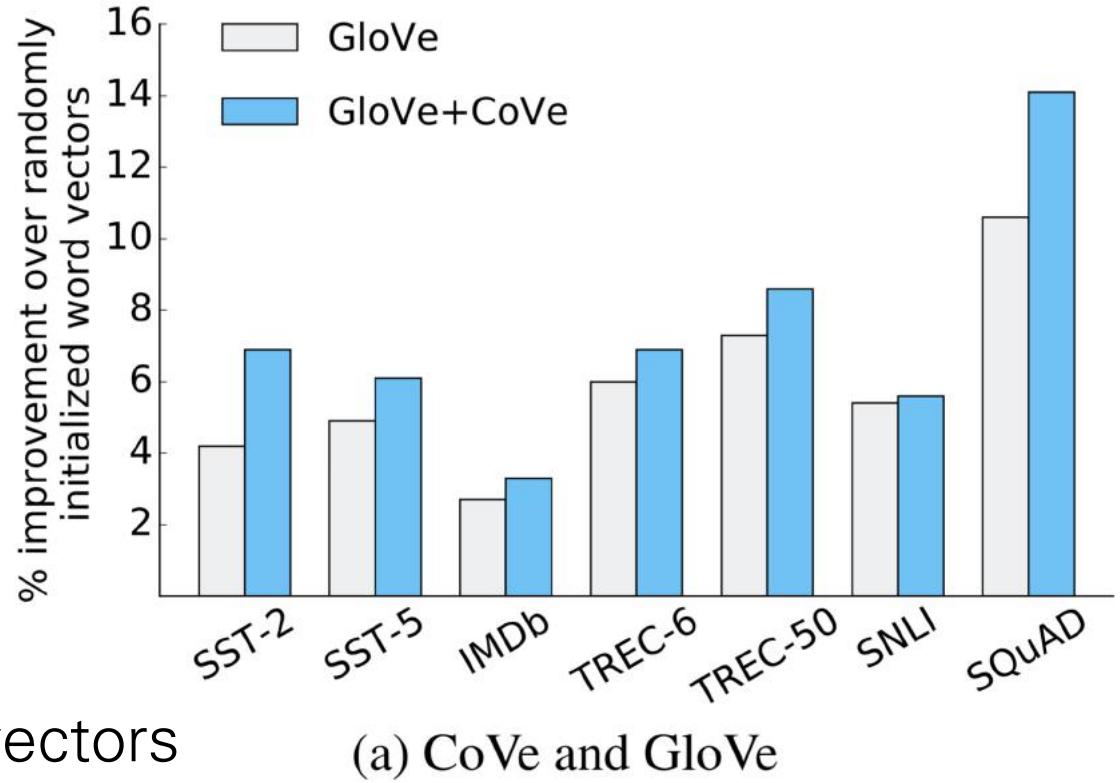
- Word vectors are the classic approach!
- GLoVE (Pennington et al. 2014)
  - Common Crawl (a good chunk of the internet)
  - Represent co-occurrences of words in 840 billion tokens



(a) CoVe and GloVe

Information and Representation Engineering alongside Architecture Engineering

- Word vectors are the classic approach!
- GLoVE (Pennington et al. 2014)
  - Common Crawl (a good chunk of the internet)
  - Represent co-occurrences of words in 840 billion tokens



(a) CoVe and GloVe

The NLI models were already using word vectors
So this hasn't been figured out yet!
But GLoVe -> ELMo -> GPT-1 -> BERT helps a ton!

Information Engineering Taking Off (CoVe, ELMo, ULMFiT, GPT-1, BERT)

- GPT-1 performs similarly to KIM (83.75%) on the new NLI test set
- BERT is basically SOTA on everything
- It's just a "stock" transformer!
- But it makes up for this with all that its learned through pre-training.

Instead of manually specifying what to predict through the creation of large supervised datasets…

Figure out how to learn from and predict everything “out there”.

You can think of everytime we build a dataset as setting the importance of everything else in the world to 0 and the importance of everything in the dataset to 1.

Our poor models! They know so little and yet still have so much hidden from them.

A Potential Recipe

1. High capacity and flexible model classes

+

2. Algos for extracting information and learning the structure of domains

+

3. An almost infeasible amount of data tiling everything (billions of unlabeled examples?)

+

4. An offensive amount of compute with which to learn

= ?

A Potential Recipe

1. High capacity and flexible model classes

+

2. Algos for extracting information and learning the structure of domains

+

3. An almost infeasible amount of data tiling everything (billions of unlabeled examples?)

+

4. An offensive amount of compute with which to learn (peta to exaflops?)

=

Is it time to stop? To call it quits?

A Potential Recipe

1. High capacity and flexible model classes

+

2. Algos for extracting information and learning the structure of domains

+

3. An almost infeasible amount of data (billions of unlabeled examples?)

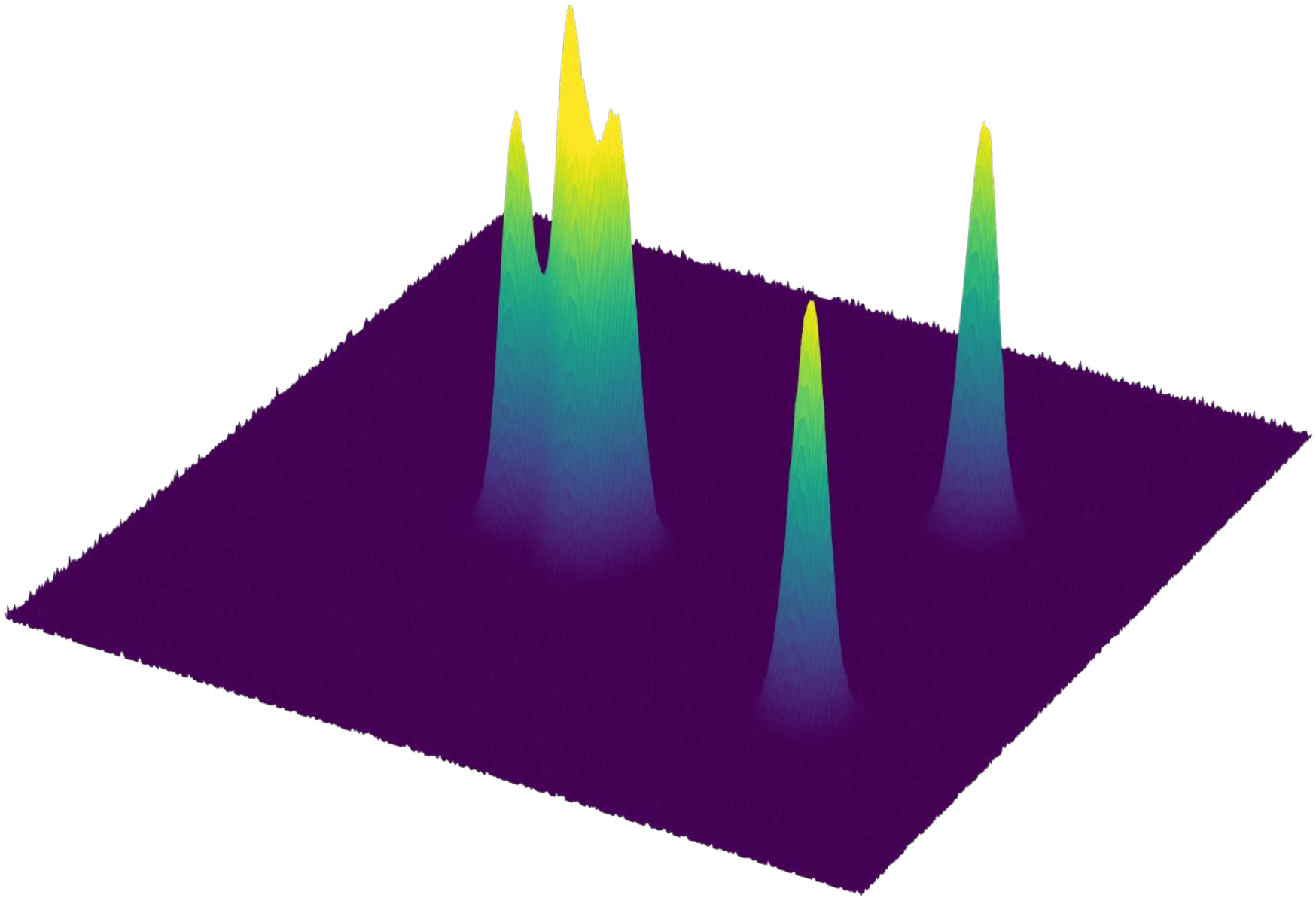
+

4. An offensive amount of compute with which to learn (peta to exaflops?)

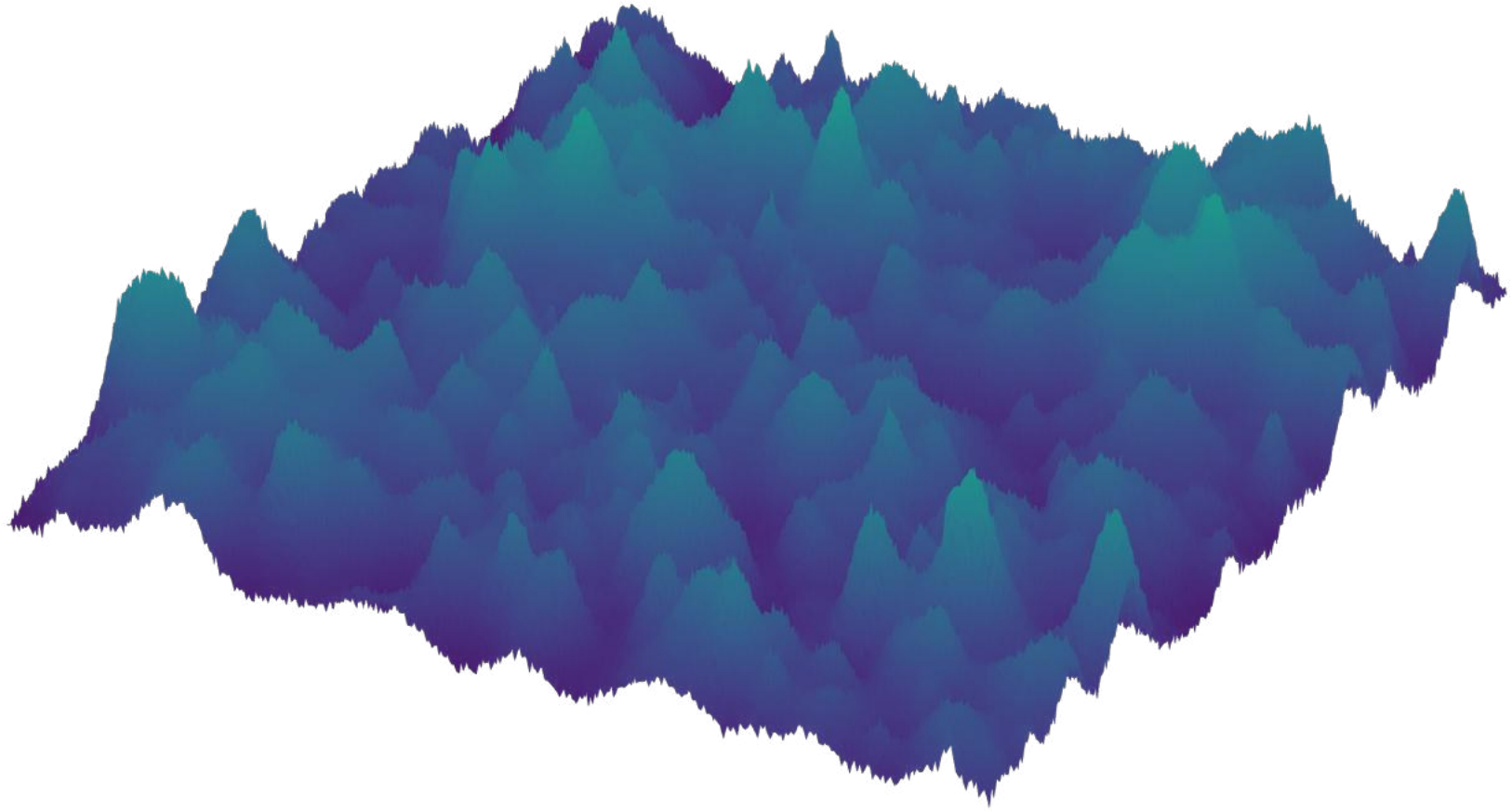
=

Or will it drive a good chunk of progress over the next few years?

What about Multitask Learning?



GPT-2?



GPT-2

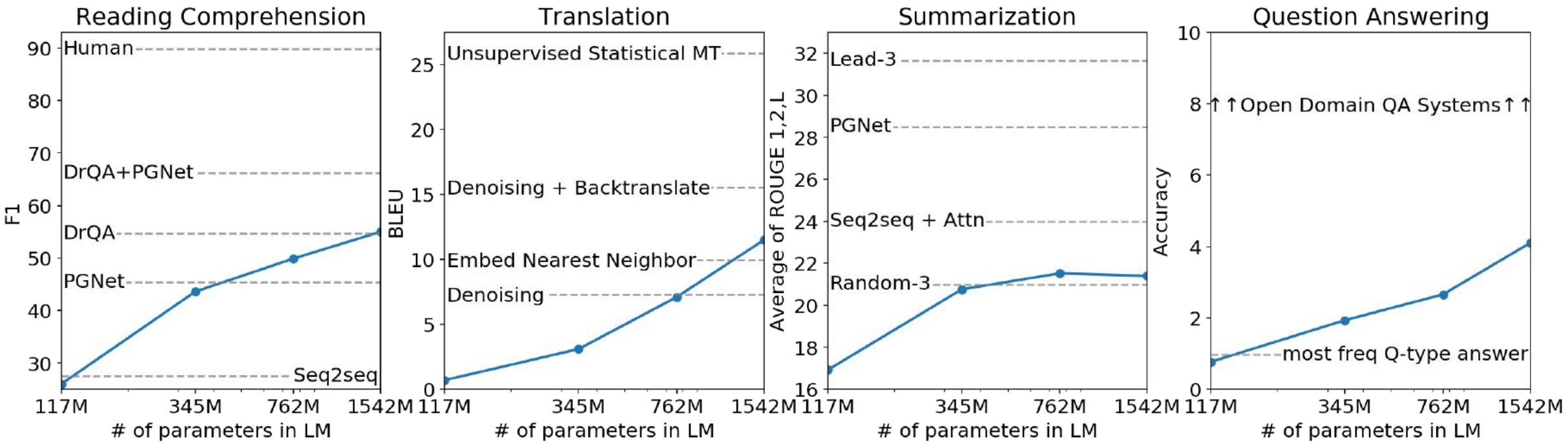
- More data
  - 40GB of text
  - 10B tokens
  - 8 million webpages
- Bigger model
  - Up to 1.5 billion parameters
  - 1024 token context
  - 48 layers, 1600 dim state

Just a language model - predicts everything (with some unfortunate restrictions as BERT shows)

Performance across tasks

DATASET	METRIC	OUR RESULT	PREVIOUS RECORD	HUMAN
Winograd Schema Challenge	accuracy (+)	**70.70%**	63.7%	92%+
LAMBADA	accuracy (+)	**63.24%**	59.23%	95%+
LAMBADA	perplexity (-)	**8.6**	99	~1-2
Children's Book Test Common Nouns (validation accuracy)	accuracy (+)	**93.30%**	85.7%	96%
Children's Book Test Named Entities (validation accuracy)	accuracy (+)	**89.05%**	82.3%	92%
Penn Tree Bank	perplexity (-)	**35.76**	46.54	unknown
WikiText-2	perplexity (-)	**18.34**	39.14	unknown
enwik8	bits per character (-)	**0.93**	0.99	unknown
text8	bits per character (-)	**0.98**	1.08	unknown
WikiText-103	perplexity (-)	**17.48**	18.3	unknown

Performance across tasks



Why it's working?

"I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I'm not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "**Mentez mentez, il en restera toujours quelque chose**," which translates as, "**Lie lie and something will always remain.**"

"I hate the word '**perfume**,"' Burr says. 'It's somewhat better in French: '**parfum**.'

If listened carefully at 29:55, a conversation can be heard between two guys in French: "**-Comment on fait pour aller de l'autre coté? -Quel autre coté?**", which means "**- How do you get to the other side? - What side?**".

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

"**Brevet Sans Garantie Du Gouvernement**", translated to English: "**Patented without government warranty**".

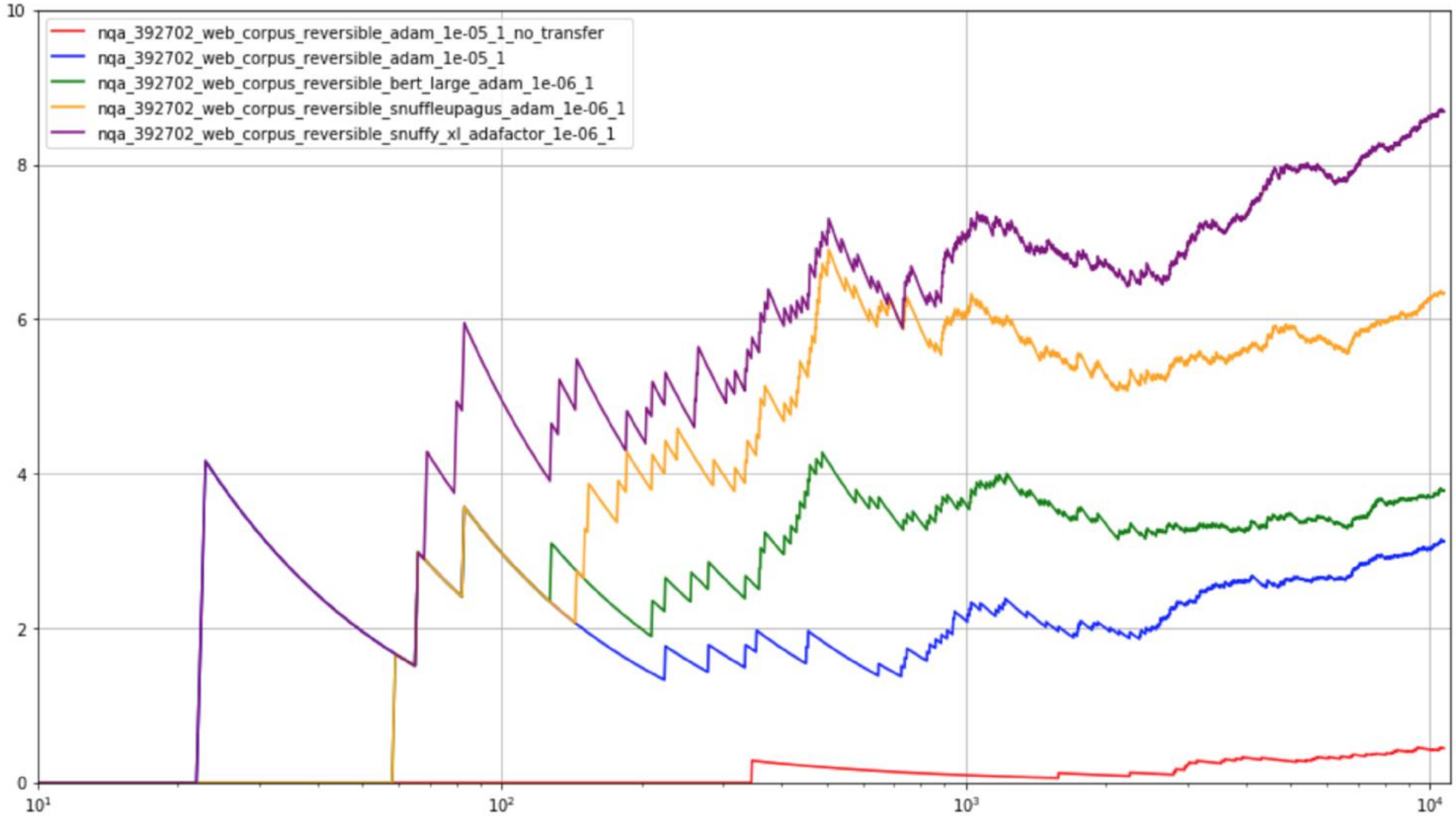
Why it's working?

Question Answering and Reading Comprehension:
6 Million 5 Ws questions in the dataset

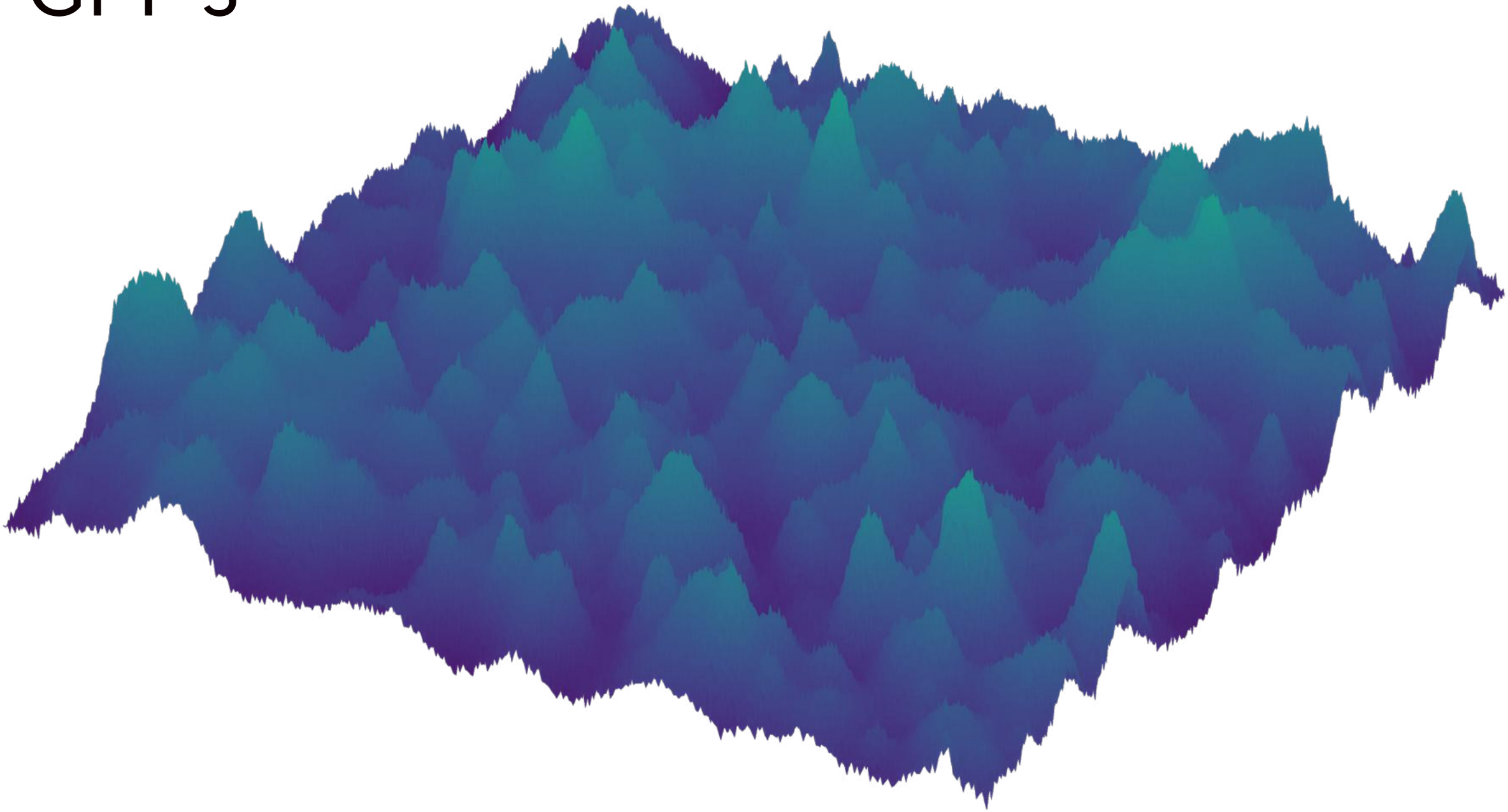
Summarization:
~100K TL;DR, In summary…

Translation:
~10MB French data

A concrete example of why unsupervised learning is necessary



GPT-3



GPT-3

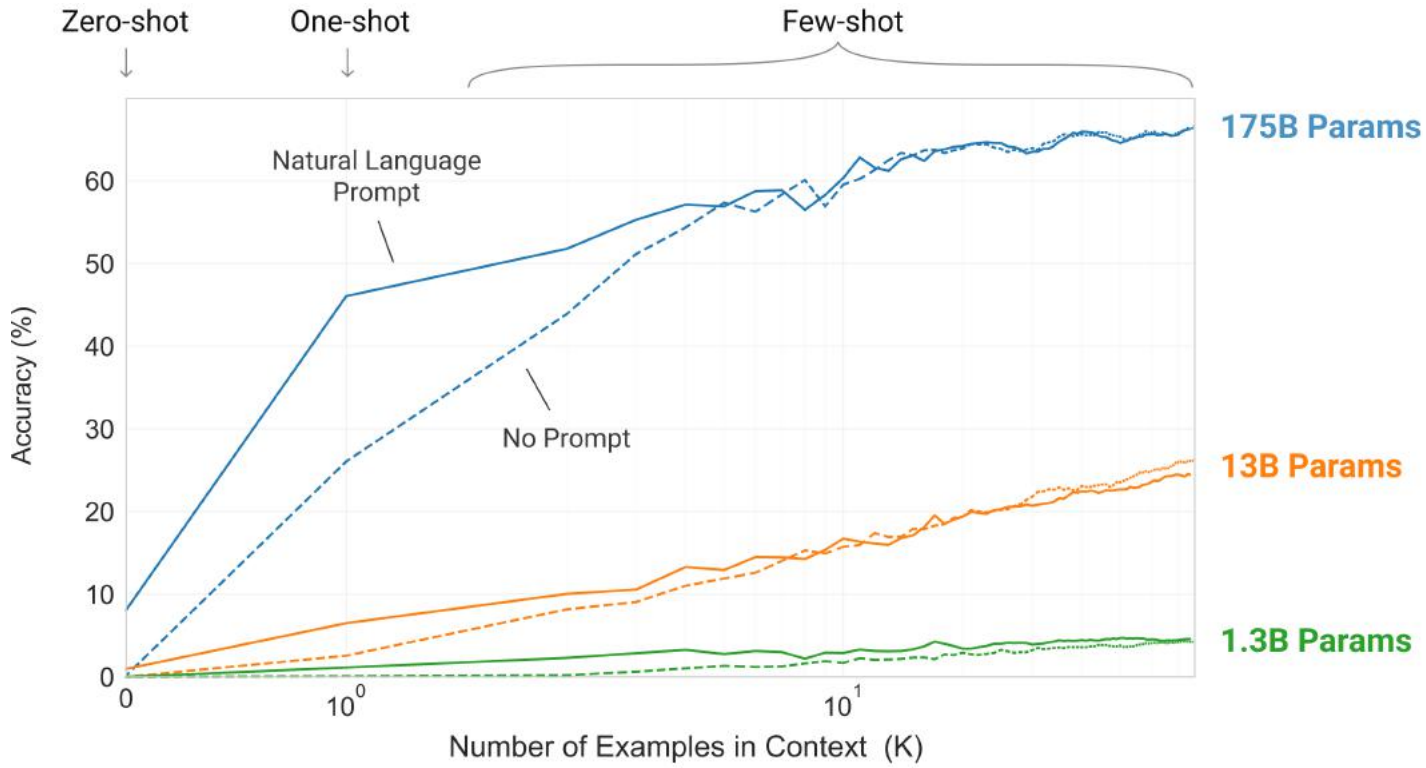


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

GPT-3

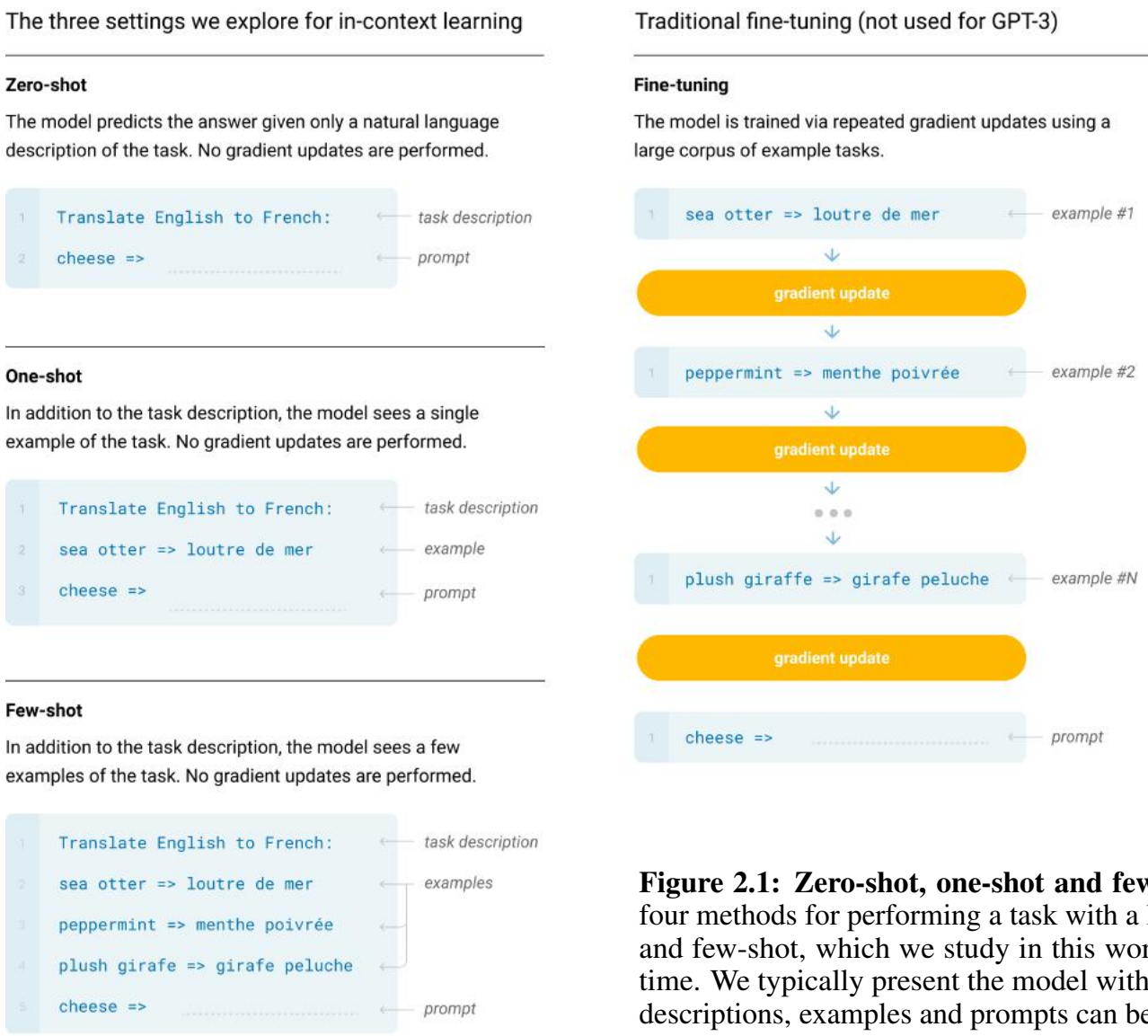


Figure 2.1: Zero-shot, one-shot and few-shot, contrasted with traditional fine-tuning. The panels above show four methods for performing a task with a language model – fine-tuning is the traditional method, whereas zero-, one-, and few-shot, which we study in this work, require the model to perform the task with only forward passes at test time. We typically present the model with a few dozen examples in the few shot setting. Exact phrasings for all task descriptions, examples and prompts can be found in Appendix G.

GPT-3

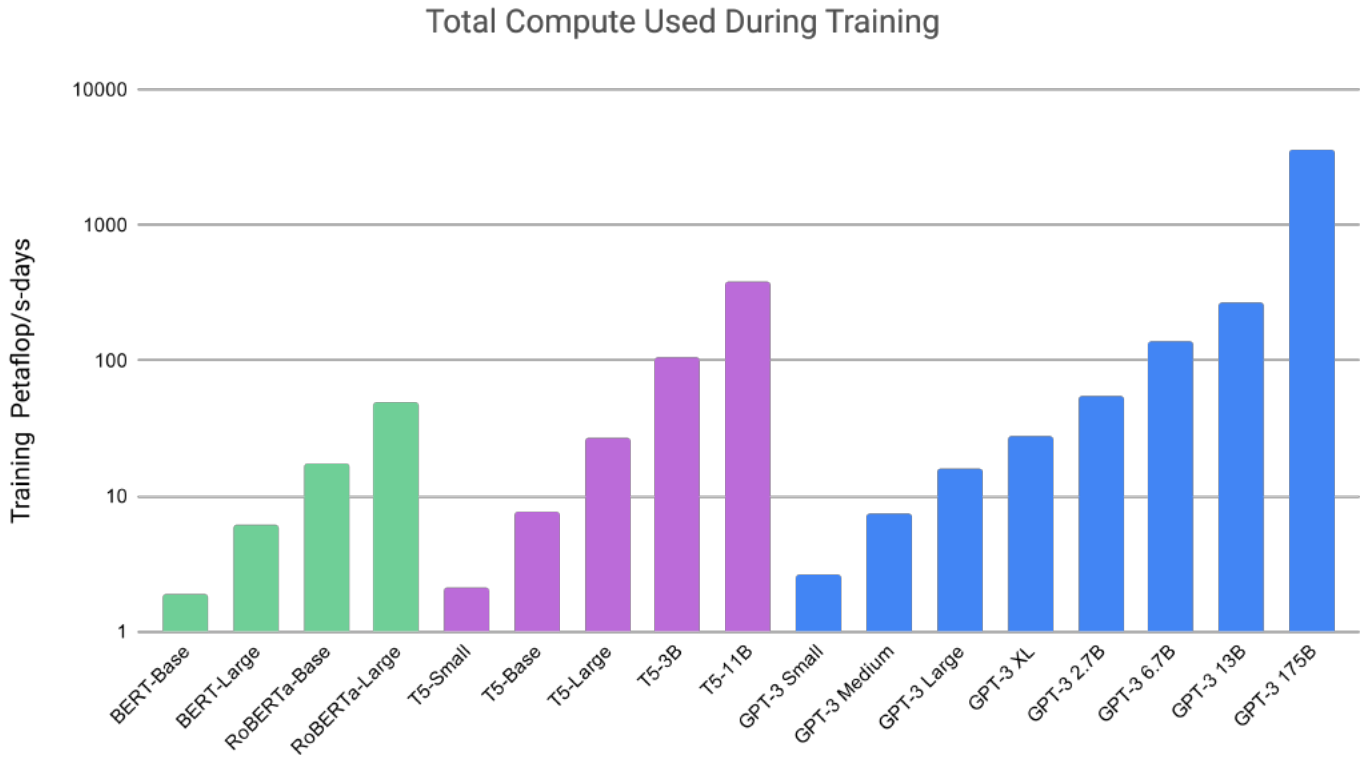


Figure 7.2: Total compute used during training. Based on the analysis in Scaling Laws For Neural Language Models [KMH+20] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly 50 petaflop/s-days of compute during pre-training. Methodology for these calculations can be found in the Appendix.

GPT-3

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: **After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.**
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

The GPT-3 generated news article that humans had the greatest difficulty distinguishing from a human written article

GPT-3

Context →	The City BY C. P. CAVAFY TRANSLATED BY EDMUND KEELEY [Poem text omitted] SOME TREES John Ashbery [Poem text omitted] Shadows on the Way Wallace Stevens

Nobody will come to this place. It is a road that leads nowhere.
The solitude is deep. The mountains are high.
But they are desolate, and they turn the traveler's face
Towards the North. All the sounds of the world are far away.
When the wind rises above the trees,
The boughs bow to the ground.
Even the birds that inhabit the tangle of weeds
That is the roadside cover, are silent. One listens,
But hears no roar of the forest. One is alone.
One will be taken.
One will be taken.
There is no utterance, there is no conversation,
But one is uneasy all the same....
There is a thin blue mist,
A darkness rising like smoke,
And within that darkness
A possession of the heart.
One will be taken.... It was here, and it will be here again-
Here, under this sky empty and full of light.

Completion from a context suggesting the model compose a poem in the style of Wallace Stevens with the title 'Shadows on the Way'.

GPT-3

Context →	Helsinki is the capital and largest city of Finland. It is in the region of Uusimaa, in southern Finland, on the shore of the Gulf of Finland. Helsinki has a population of , an urban population of , and a metropolitan population of over 1.4 million, making it the most populous municipality and urban area in Finland. Helsinki is some north of Tallinn, Estonia, east of Stockholm, Sweden, and west of Saint Petersburg, Russia. Helsinki has close historical connections with these three cities. The Helsinki metropolitan area includes the urban core of Helsinki, Espoo, Vantaa, Kauniainen, and surrounding commuter towns. It is the world's northernmost metro area of over one million people, and the city is the northernmost capital of an EU member state. The Helsinki metropolitan area is the third largest metropolitan area in the Nordic countries after Stockholm and Copenhagen, and the City of Helsinki is the third largest after Stockholm and Oslo. Helsinki is Finland's major political, educational, financial, cultural, and research center as well as one of northern Europe's major cities. Approximately 75% of foreign companies that operate in Finland have settled in the Helsinki region. The nearby municipality of Vantaa is the location of Helsinki Airport, with frequent service to various destinations in Europe and Asia. Q: what is the most populous municipality in Finland? A: Helsinki Q: how many people live there? A: 1.4 million in the metropolitan area Q: what percent of the foreign companies that operate in Finland are in Helsinki? A: 75% Q: what towns are a part of the metropolitan area? A:
Target Completion →	Helsinki, Espoo, Vantaa, Kauniainen, and surrounding commuter towns

Formatted dataset example for CoQA

Takeaways from scaling language modeling

- Performance not (usually) limited by something a single paper fixes
- Diminishing returns mean there is always some other bottleneck
  - Fancy model -> compute utilization, trainability
  - Parameters -> compute
  - Data -> capacity
  - Capacity -> data, compute

- **Be pragmatic about scaling**
- If you do everything sensibly - compute will probably be the bottleneck
  - If it's not… there's an interesting research problem!

How to do research on large scale models?

How to do research on large scale models?

- Don’t do research on large scale models

How to do research on large scale models?

- Don’t do research on large scale models
- Prototype on models which are 10x smaller and 10x faster
  - Run 10x as many experiments in parallel instead
  - Every behavior in the GPT-2 paper shows up on these models

- After the proof of concept - then you scale
  - GPT-1 was a proof point on zero-shot task transfer
  - GPT-1 on WebText is already SOTA on several LM tasks

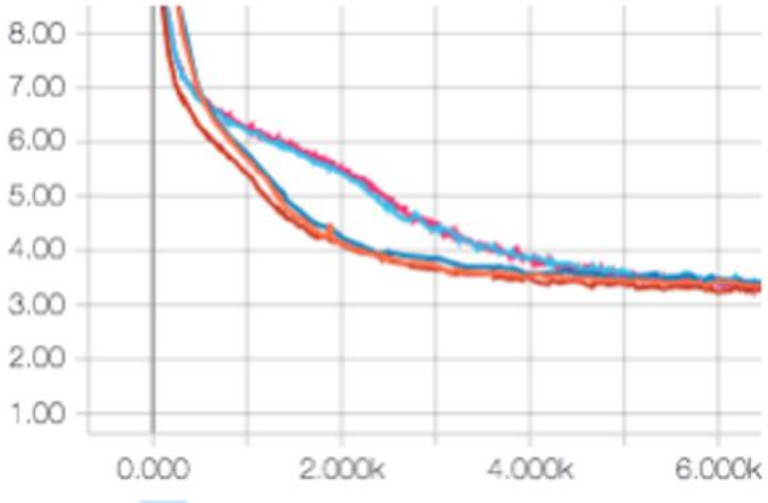
- Used the same strategy for Sentiment Unit
  - First trained a 512 dim LSTM in a few days
  - Final 4096 dim LSTM took a month

How to do research on large scale models?

- Develop and test everything quickly at small scale first
- Tune the hyperparams, decide on a model, checkout datasets, etc...
- Whatever does best at a reasonable scale will also probably do well at large scale
- Optimize the language model as a language model
  - Log-prob of held-out text
  - Then see what it else it can do

The Gotcha's

- Sometimes issues don't show up until at scale
- Plan for something to break about every order of magnitude of scale
- Will have to re-tune hyperparameters
- For GPT-2 models this happened at >= 24 self-attention blocks
  - Performance of models appears to saturate
  - Fix was better weight init and pre-activation style residual network
    - Rewon Child figured this out



More Model More Problem

- Self-attention architectures + long sequences = **lots of memory**
- Recompute
- Half Precision (FP-16)
- Data Parallelism

Write Efficient / Smart Code!

- Naive tensorflow code can now be over 5 times slower than what is achievable on modern hardware

Case study: GPT-1

- Took 25 days on 8 P6000s (how do you do research on models that take a month to train without going insane?)

- Trains in 3 days on 8 v100s
  - 1.75x from TF data parallel -> MPI + NCCL AllReduce
  - 1.50x from native TF ops -> Blocksparse ops
  - 3.50x from FP32 Pascal -> FP16 Volta

Blocksparse Library - Scott Gray

- Accelerated primitives for common Tensorflow ops
  - Dropout, normalization, optimizers, activations
- Custom self-attention operations
  - Avoid transposes, fuse operations, sparse compute
- Targets Volta / Turing hardware
  - Tensorcores allow for 3x+ speedup over previous gen hardware

```
from blocksparse.transformer import BlocksparseTransformer, softmax_cross_entropy
from blocksparse.optimize    import AdamOptimizer, ClipGlobalNorm
from blocksparse.norms       import layer_norm
from blocksparse.embed       import embedding_lookup
from blocksparse.ewops       import bias_relu, dropout
from blocksparse.nccl        import allreduce
```

What is the Sweet Spot in terms of compute?

- If you're paying for it:
  - A 4 2080 Ti desktop
    - The results in GPT-2 do show up on models trainable on this hardware (but will take a week)
    - Can ~ match BERT-Base in that time too
    - Cost about $6,000 :(
- If someone else is paying for it:
  - 8 v100s from a cloud provider (AWS, GCE, etc...)

Takeaways from language modeling

- Scale matters go beyond classic datasts like PTB
- Better results come from combining several sources of improvement
- Don’t get bottlenecked by something that can be fixed easily
- Don’t let scale slow you down during development
- A medium+ language model on a new dataset / domain will probably learn something interesting - but might take some digging to find
  - Most of my research for the past few years has been exploring the capabilities, behaviors, and uses of language models in this regime

Where is this Heading?

- In the next few years language models will be trained on pretty much the whole internet (might as well throw in millions of books too!)
- Will scaling trends breakdown?
- How far will this get?

Where is this Heading?

- In the next few years language models will be trained on pretty much the whole internet (might as well throw in millions of books too!)
- Will scaling trends breakdown?
- How far will this get?
- If trendlines continue…

Where is this Heading?

- In the next few years language models will be trained on pretty much the whole internet (might as well throw in millions of books too!)
- Will scaling trends breakdown?
- How far will this get?
- If trendlines continue…

It will probably feel unsatisfying, though