Lecture 11: NLG, BART, GPT2, GPT3

Andrei Arsene Simion

Columbia University

April 12, 2023

Outline

- Natural Language Degeneration
- GPT2
- GPT3
- SpanBERT
- BART
- ALBERT

Outline

- Natural Language Degeneration
- GPT2
- GPT3
- SpanBERT
- BART
- ALBERT

NLG

- Once we have a model, how do we sample from it?
- We saw two ways: Greedy and Beam Search decoding
- Observation: maximization-based decoding methods such as Beam Search lead to degeneration: you get text that is bland, incoherent, or gets stuck in repetitive loops
- On the other hand, random sampling is incoherent

Context: 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.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

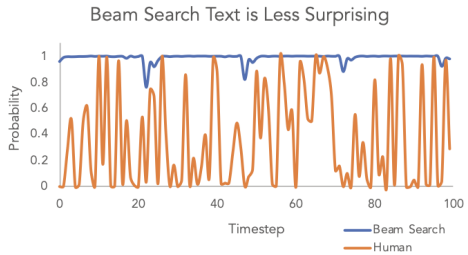
Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

Figure 1: Even with substantial human context and the powerful GPT-2 Large language model, Beam Search (size 32) leads to degenerate repetition (highlighted in blue) while pure sampling leads to incoherent gibberish (highlighted in red). When $b \geq 64$, both GPT-2 Large and XL (774M and 1542M parameters, respectively) prefer to stop generating immediately after the given context.

- When GPT2 came out, it sampled a very real looking piece of text but did not use Beam Search, it used another decoding method called Top-K sampling
- Top-K sampling still has problems, but its good to look at some alteratives

Problems with Decoding: Another Example

- Human speech is not greedy-maximization based
  - For human text, a token's probability goes up and down, we don't always pick high probability tokens
  - Given the above, Beam Search is unnatural



Beam Search

...to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and...

Human

...which grant increased life span and three years warranty. The Antec HCG series consists of five models with capacities spanning from 400W to 900W. Here we should note that we have already tested the HCG-620 in a previous review and were quite satisfied With its performance. In today's review we will rigorously test the Antec HCG-520, which as its model number implies, has 520W capacity and contrary to Antec's strong beliefs in multi-rail PSUs is equipped...

- On the other hand, random sampling has another problem: the "unreliable tail"
  - There are several tokens with small probabilities that should not be sampled, but are given a chance and result in gibberish
- Eval: Pick generated text and compare the probability of the generated words to some reference, such as human text

Top K Sampling

- One direct way to remove the tail is Top-K sampling
- This is easy to explain: at each time step, consider only the K tokens with the highest probabilities
  - When decoding at time t, select a set V_K^t of K top tokens
  - Set $q = \sum_{x \in V_K^t} P(x|x_0, \ldots, x_{t-1})$
  - Renormalize and sample using $Q(x|x_0, \ldots, x_{t-1}) = P(x|x_0, \ldots, x_{t-1})/q$
- Good: This method removes tokens with a low probability, so this is good because it truncates the tail
- Problem: What is K? Should K always be the same?
- Problem: To match human text, we need K large; this leads to incoherence
- Problem: I.e. You *need* low probability words sometime
- Problem: q above varies greatly at each time step, hard to understand the gears

Temperature Scaling

- Fix $T > 0$
- Consider the token distribution p but this time work with $q = p/T$
- Let $T \to 0$
- As we saw, this has the effect of pushing tokens with high probabilities higher, and suppressing tokens with lower probabilities
- Problem: Empirically, this type of distribution "truncation" still leads to incoherence
- Problem: While lowering the temperature improves generation quality, it comes at the cost of decreasing diversity

Nucleus Sampling (2020)

- Fix $p \in (0, 1)$
- For example, $p = 0.95$
- At time t, select V_p^t such that

$$q = \sum_{x \in V_p^t} P(x|x_0, \ldots, x_{t-1}) > p$$

- Note that at each time step we are selecting a dynamic set of tokens
- Once the set V_p^t is selected, renormalize the token probabilities so that

$$Q(x|x_0, \ldots, x_{t-1}) = P(x|x_0, \ldots, x_{t-1})/q$$

- Sample from Q

Results

- More examples of incoherence and gibberish

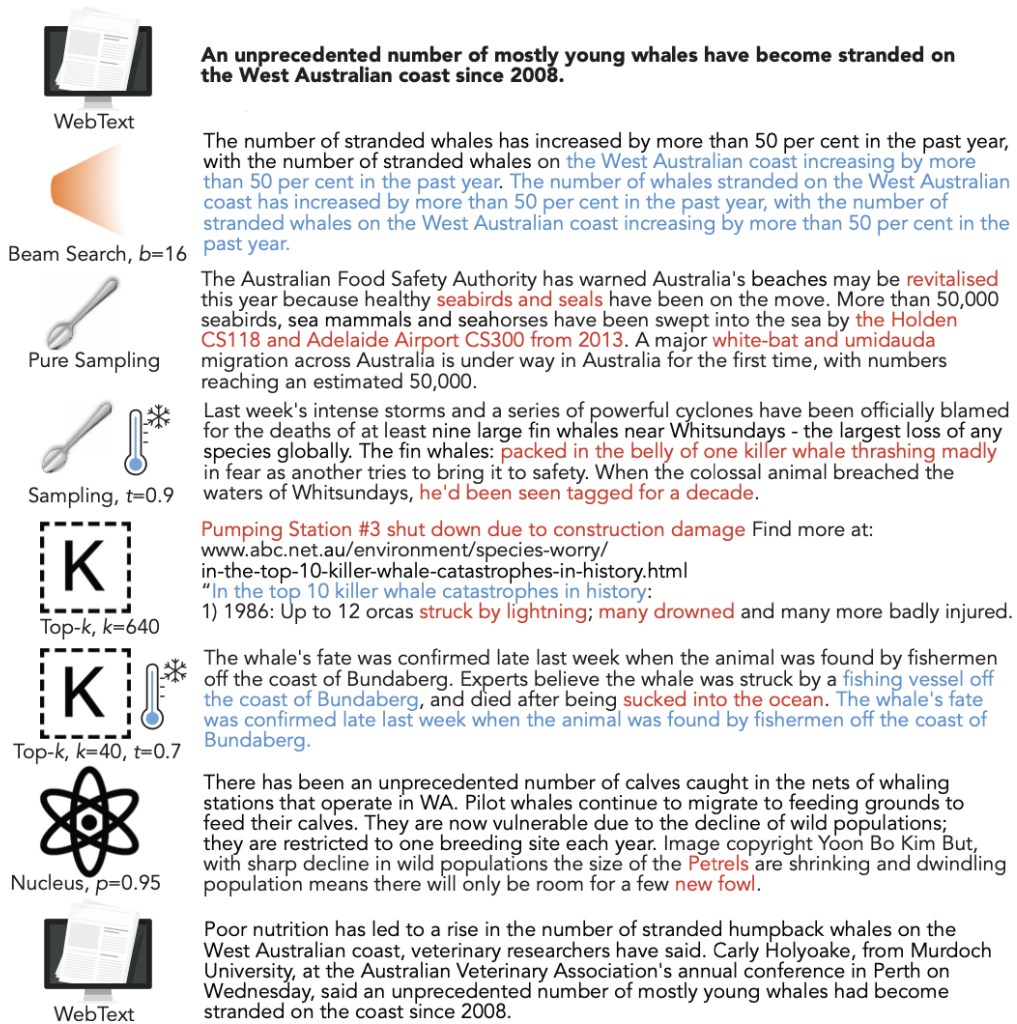


Figure 3: Example generations continuing an initial sentence. Maximization and top-k truncation methods lead to copious repetition (highlighted in blue), while sampling with and without temperature tends to lead to incoherence (highlighted in red). Nucleus Sampling largely avoids both issues.

Repetition is rewarded when decoding

- Repetition is rewarded when decoding greedily

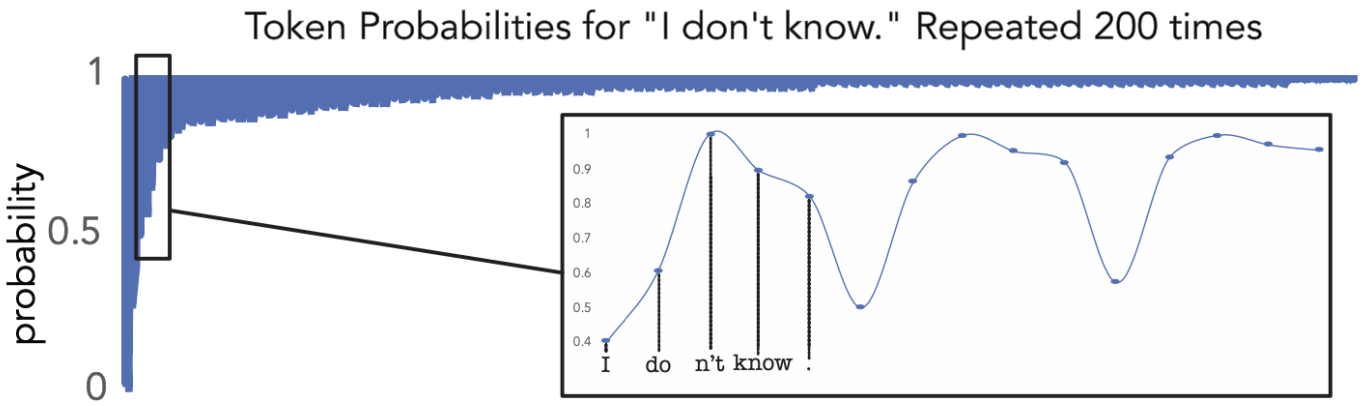


Figure 4: The probability of a repeated phrase increases with each repetition, creating a positive feedback loop. We found this effect to hold for the vast majority of phrases we tested, regardless of phrase length or if the phrases were sampled randomly rather than taken from human text.

Perplexity and Top-K issues

- Using just lower perplexity is not a good idea: Beam Search wins but it's not ideal

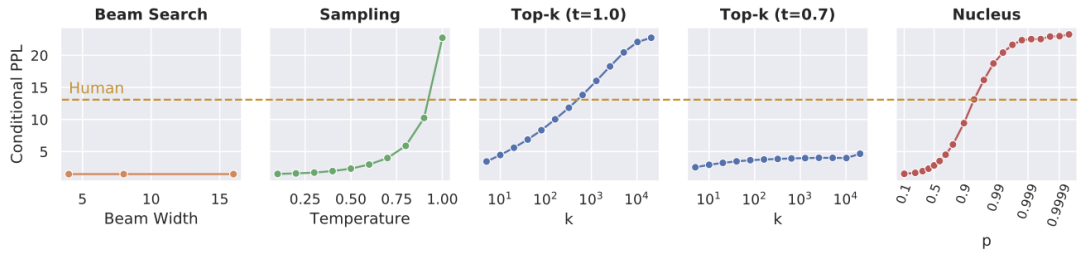


Figure 6: Perplexities of generations from various decoding methods. Note that beam search has unnaturally low perplexities. A similar effect is seen using a temperature of 0.7 with top-k as in both Radford et al. (2019) and Fan et al. (2018). Sampling, Top-k, and Nucleus can all be calibrated to human perplexities, but the first two face coherency issues when their parameters are set this high.

- Top K sampling can't deal well with flat or peaked distributions - you need a different K each time!

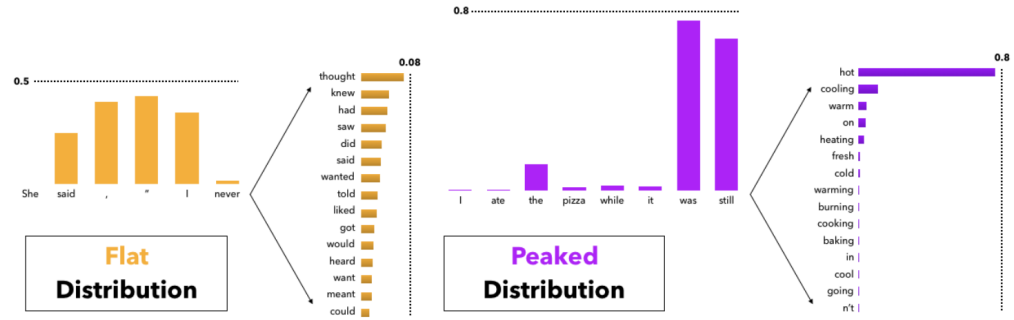


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

Results

- Beam Search suffers the most from repetition

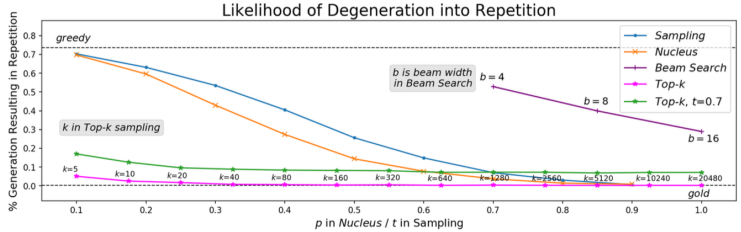


Figure 9: We visualize how often different decoding methods get "stuck" in loops within the first 200 tokens. A phrase (minimum length 2) is considered a repetition when it repeats at least **three** times at the *end* of the generation. We label points with their parameter values except for t and p which follow the x-axis. Values of k greater than 100 are rarely used in practice and values of p are usually in $[0.9, 1)$; therefore Nucleus Sampling is far closer to the human distribution in its usual parameter range. Sampling with temperatures lower than 0.9 severely increase repetition. Finally, although beam search becomes less repetitive according to this metric as beam width increases, this is largely because average length gets shorter as b increases (see Appendix A).

- Overall Nucleus Sampling, gave perplexities closest to Human text, there are lots of other directions

Method	Perplexity	Self-BLEU4	Zipf Coefficient	Repetition %	HUSE
Human	12.38	0.31	0.93	0.28	-
Greedy	1.50	0.50	1.00	73.66	-
Beam, b=16	1.48	0.44	0.94	28.94	-
Stochastic Beam, b=16	19.20	0.28	0.91	0.32	-
Pure Sampling	22.73	0.28	**0.93**	0.22	0.67
Sampling, t=0.9	10.25	0.35	0.96	0.66	0.79
Top-k=40	6.88	0.39	0.96	0.78	0.19
Top-k=640	13.82	**0.32**	0.96	**0.28**	0.94
Top-k=40, t=0.7	3.48	0.44	1.00	8.86	0.08
Nucleus p=0.95	**13.13**	**0.32**	0.95	0.36	**0.97**

Table 1: Main results for comparing all decoding methods with selected parameters of each method. The numbers *closest to human scores* are in **bold** except for HUSE (Hashimoto et al., 2019), a combined human and statistical evaluation, where the highest (best) value is **bolded**. For Top-k and Nucleus Sampling, HUSE is computed with interpolation rather than truncation (see §6.1).

Outline

- Natural Language Degeneration
- GPT2
- GPT3
- SpanBERT
- BART
- ALBERT

GPT2

- We saw Encoder-Decoder Transformers can be used to get SOTA Machine Translation results
- By training to predict the next word, we also saw GPT-1 can be used generate text and do other tasks (via fine tuning)
- What happens if you make the model larger?
- What is "inside" of GPT2 that we can use directly, without fine tuning?

GPT2 - The first large Decoder

- How much larger was GPT2 vs GPT1? What are the differences?
  - 100 Million parameters $\rightarrow$ 1.5 Billion Parameters
  - Small rearranging of layer norm and residual layers, another layer norm
    - $LN(x + FF(x)) \rightarrow x + FF(LN(x))$
    - $MHA(x) \rightarrow MHA(LN(x))$
  - Increase vocabulary from 30 K to 50 K
  - Increase context window T from 512 to 1024
- Trained on WebText, consisting of crawls of 45 Million links
- Specifically, the authors show that a language model can learn general tasks without any fine tuning
- The authors believe that a large enough language model will begin to learn the tasks embedded within the natural language itself and won't require any additional supervision
- What are some examples?

GPT2 - Example Decoding

- Decoding (via Top K) with GPT2 allows us to generate text that is very "real" sounding
- For its time (2019), this was extraordinary

Context (human-written): 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.

GPT-2: 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 Pérez, 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. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez 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 Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

GPT2 - General Results

- For GPT2 the authors wanted to see what a model learns both generally and what it can generalize to
- They did this by augmenting the data and seeing what the model would generate - no gradient updates!
  - Machine Translation: (english to french, english sentence, french sentence)
  - Question Answering: (answer the question, document, question, answer)
  - Summarization: (text tl;dr)

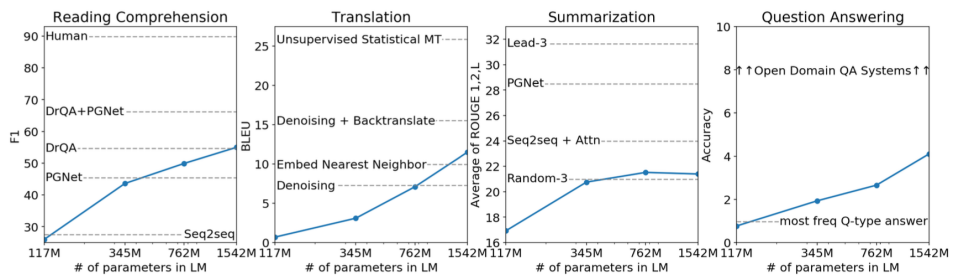


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

GPT2 - Language Modeling

- Evaluated GPT2 on several datasets where the task is loosely speaking based on predicting language
- LAMBADA

Context: The battery on Logan's radio must have been on the way out. So he told himself. There was no other explanation beyond Cygan and the staff at the White House having been overrun. Lizzie opened her eyes with a flutter. They had been on the icy road for an hour without incident.
Target sentence: Jack was happy to do all of the _____.
Target word: driving

- Children's Book Corpus

```
Context:
1 So they had to fall a long way .
2 So they got their tails fast in their mouths .
3 So they could n't get them out again .
4 That 's all .
5 `` Thank you , " said Alice , `` it 's very interesting .
6 I never knew so much about a whiting before . "
7 `` I can tell you more than that , if you like , " said the Gryphon .
8 `` Do you know why it 's called a whiting ? "
9 `` I never thought about it , " said Alice .
10 `` Why ? "
11 `` IT DOES THE BOOTS AND SHOES . '
12 the Gryphon replied very solemnly .
13 Alice was thoroughly puzzled .
14 `` Does the boots and shoes ! "
15 she repeated in a wondering tone .
16 `` Why , what are YOUR shoes done with ? "
17 said the Gryphon .
18 `` I mean , what makes them so shiny ? "
19 Alice looked down at them , and considered a little before she gave her answer .
20 `` They 're done with blacking , I believe . "

Query: `` Boots and shoes under the sea , " the XXXXX went on in a deep voice , `` are done

Candidates: Alice|BOOTS|Gryphon|SHOES|answer|fall|mouths|tone|way|whiting

Answer: gryphon
```

GPT2 - Language Modeling

- For LM, GPT2 was SOTA on several baselines

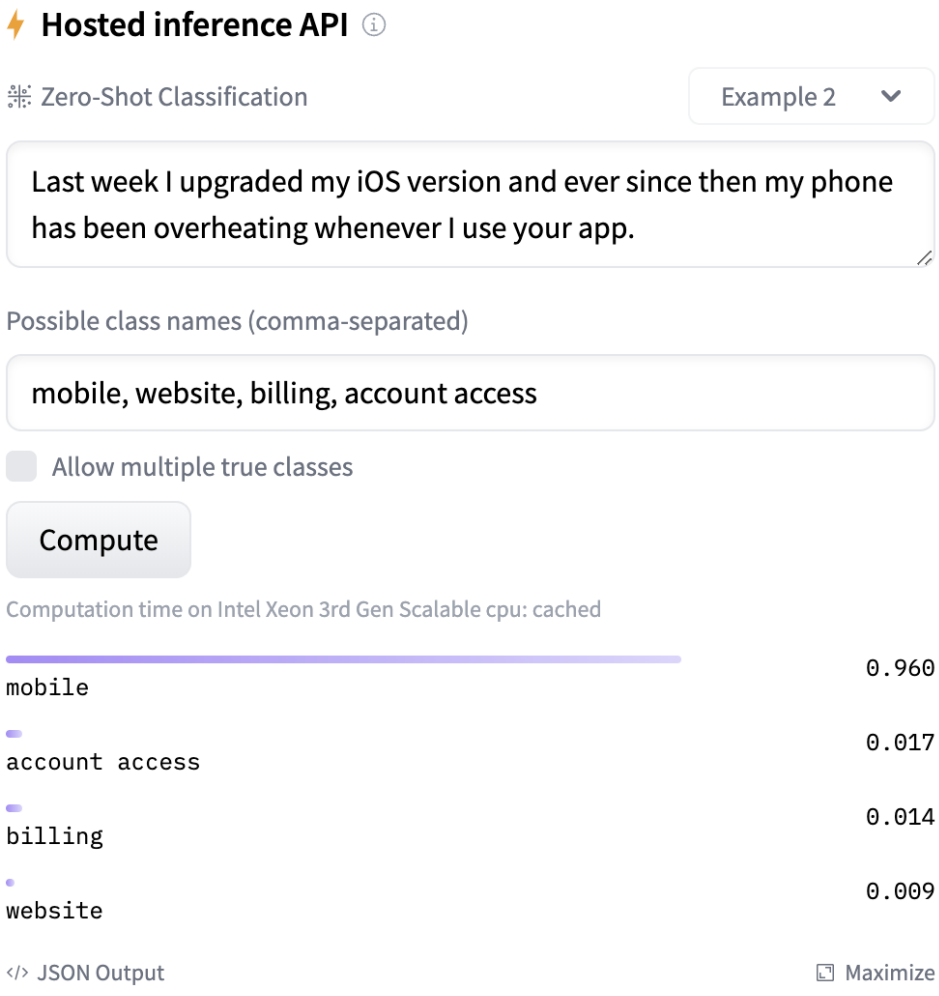
Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

GPT2 - Zero Shot Learning

- Zero Shot Example on Hugging Face



GPT2 - Zero Shot Learning

Winograd Schema

5. **SAME TEXT AS PREVIOUS QUESTION**
   Babar wonders how he can get new clothing. Luckily, a very rich old man who has always been fond of little elephants understands right away that he is longing for a fine suit. As he likes to make people happy, he gives **him** his wallet.

   Snippet: gives **him**

   A. Babar
   B. old man

6. Dan had to stop Bill from toying with the injured bird. **He** is very compassionate.

   Snippet: **He** is very compassionate.

   A. Dan
   B. Bill

7. **NOTE: TEXT IS DIFFERENT FROM PREVIOUS QUESTION**
   Dan tried to stop Bill from getting help for the injured bird. **He** is very callous.

   Snippet: **He** is very callous.

   A. Dan
   B. Bill

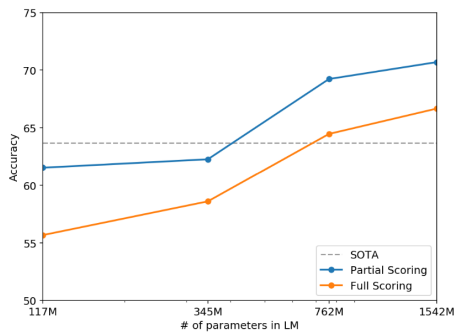


Figure 3. Performance on the Winograd Schema Challenge as a function of model capacity.

GPT2 - Other Tasks

- Reading Comprehension - not terrible but it seems the model fills in any "name" to *who* questions and has similar patterns to other types of questions
- Summarization - Add "tl:dr" to context; not great

	R-1	R-2	R-L	R-AVG
Bottom-Up Sum	**41.22**	**18.68**	**38.34**	**32.75**
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 `TL;DR:`	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

Table 4. Summarization performance as measured by ROUGE F1 metrics on the CNN and Daily Mail dataset. Bottom-Up Sum is the SOTA model from (Gehrmann et al., 2018)

- Translation - BLEU was 5 which was not great
- Question Answering; poor on SQUAD, better on "generation"

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%
Who plays ser davos in game of thrones?	Peter Dinklage	✗	52.1%
Who appoints the chair of the federal reserve system?	Janet Yellen	✗	51.5%
State the process that divides one nucleus into two genetically identical nuclei?	mitosis	✓	50.7%
Who won the most mvp awards in the nba?	Michael Jordan	✗	50.2%
What river is associated with the city of rome?	the Tiber	✓	48.6%
Who is the first president to be impeached?	Andrew Johnson	✓	48.3%
Who is the head of the department of homeland security 2017?	John Kelly	✓	47.0%
What is the name given to the common currency to the european union?	Euro	✓	46.8%
What was the emperor name in star wars?	Palpatine	✓	46.5%
Do you have to have a gun permit to shoot at a range?	No	✓	46.4%
Who proposed evolution in 1859 as the basis of biological development?	Charles Darwin	✓	45.7%
Nuclear power plant that blew up in russia?	Chernobyl	✓	45.7%
Who played john connor in the original terminator?	Arnold Schwarzenegger	✗	45.2%

Table 5. The 30 most confident answers generated by GPT-2 on the development set of Natural Questions sorted by their probability according to GPT-2. None of these questions appear in WebText according to the procedure described in Section 4.

GPT2 - No diminishing returns yet

- The model is under fit: we can fit bigger models!

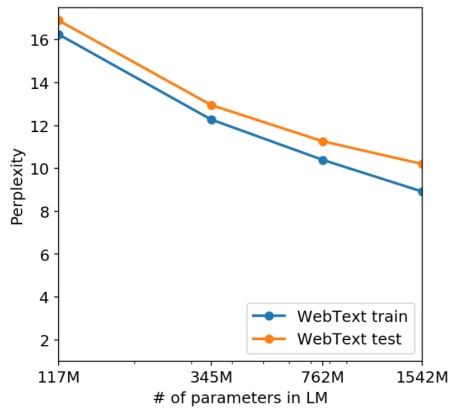


Figure 4. The performance of LMs trained on WebText as a function of model size.

- Controversial: Open AI did not release their code / model and this caused controversy upon GPT2's release
- Safety Concerns: Undetectable bots?
- Train / Test leakage? Did Test data get into the training set?

GPT2

- To test "memorization" vs "generation" the authors took new text and observed the model's completion

Context (WebText test)	
Corporal Michael P. Goeldin was an unskilled laborer from Ireland when he enlisted in Company A in November 1860. Goldein survived the war. Corporal Patrick O'Neal, also from Ireland, first enlisted in 1854 and served with Company L, 3d U.S. Artillery, in Oregon. He returned to the East Coast and enlisted in the company in 1860. O'Neal served until 1874, when he was named superintendent of the National Cemetery at Willets Point, New York. Corporal Benjamin Browne was a shoemaker from Orange County, New York. In August 1862, he enlisted in the newly formed 124th New York Volunteers, and was one of sixty-one men who transferred into Company A that October. Browne reenlisted in the company in February 1864 while it was camped at Brandy Station. He returned to civilian life after completing his enlistment in 1867. On 10 June, Artificer William Collins was promoted to corporal, probably to fill a combat leadership void for the crossing of the James River. Collins's service record does not reflect the qualities he demonstrated to earn this promotion, but he had obviously overcome some serious problems. Born in Sacketts Harbor, New York, Collins enlisted in the company in December 1853 at the age of twenty-two, and reenlisted in December 1858. Just a month before the war began in April 1861, Collins went "over the hill" and was not caught until three years later. Returned to the company on 22 March 1864, he was tried	
Small model completion	**GPT-2 completion**
on military charges and was discharged from the U.S. Military. He returned to the company, and was assigned to Division of Internal Affairs during the Civil War. As an engineer, Collins also had training in engineering from his native New York. After graduate studies in the Union College of Engineering, he was employed in the United States Army for two years at the same time. There he worked under Captain Daniel Pfeiffer and was promoted to colonel in May 1870, three years after Captain Pfeiffer's enlistment. Collins returned home to England after completing his first year in the First Division to serve the next two years in the Army Corps of Engineers, and by his second year, he began to experience his service duty.¡—endoftext—¿MOSCOW, July 5 (Reuters) - U.K.-based Russian state media and government sources and experts say Moscow has sought to downplay the "military buildup" of the Syrian opposition while maintaining support for President Donald Trump's plan to build a wall along the U.S.-Mexico border, calling it a "military coup". U.S.-led Syria war veteran and Syria envoy Michael Flynn (R), right, gestures with U.S.-based Syrian political analyst Ahmet Koryat, with	in June of that year. Sentenced to be shot for desertion, he remained on active duty. Captain James J. Dall was born in Virginia in 1829 during the Revolution and joined his father in the Army in 1836. He served with the 2d Virginia Infantry until 1845, when he went to the Western Reserve Regiment; and, on his discharge, he reenlisted with Company I, 25th New York Volunteers (which in 1845 numbered about 4,500 men). One of his brothers, James, served with his company in the same capacity in the same brigade. While the regiment remained in Connecticut through 1846, Captain Dall served with Company I at various times before his discharge. A veteran of the Mexican War, he had served two years on the staff of Brigadier General John J. Sullivan at San Antonio. During the Peninsula Campaign of 1863, he commanded a company of twenty-three and was in charge of the battery of fifty-four heavy mortars and the gunboat Pinta of the Battery, a gunboat and several other boats. Captain Dall was on active duty with Company I in the fall of 1865. Two years later, he was promoted to second lieutenant. Captain John A. O'

Outline

- Natural Language Degeneration
- GPT2
- GPT3
- SpanBERT
- BART
- ALBERT

GPT3

- For the GPT3 paper, the authors want to see how much better a model will do vs GPT2 if its larger and you provide different context
- They also want to see how adding different types of context affects the model's answers
- GPT3 vs GPT2
  - 1.5 Billion parameters $\rightarrow$ 175 Billion parameters
  - Double the context size: 1024 $\rightarrow$ 2048
  - Larger token embeddings: 1.6 K $\rightarrow$ 12.8 K
  - Attention pattern from Sparse Transformer
- Train on more data

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 used to train GPT-3. "Weight in training mix" refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets 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.

GPT3 - Hidden Learning

- The authors make the argument that while you are predicting the next word, you actually learn In-Context Learning
- I.e., you see many examples one after another that are the "same", so this context can be used to guide the model along

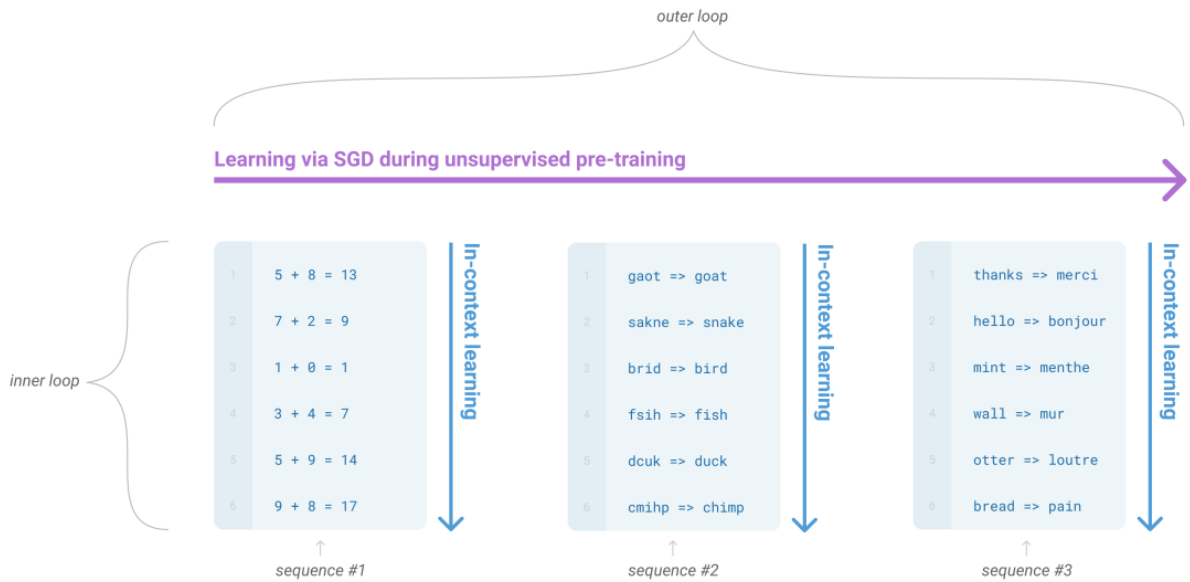


Figure 1.1: Language model meta-learning. During unsupervised pre-training, a language model develops a broad set of skills and pattern recognition abilities. It then uses these abilities at inference time to rapidly adapt to or recognize the desired task. We use the term "in-context learning" to describe the inner loop of this process, which occurs within the forward-pass upon each sequence. The sequences in this diagram are not intended to be representative of the data a model would see during pre-training, but are intended to show that there are sometimes repeated sub-tasks embedded within a single sequence.

GPT3 - Learning

- The goal of the GPT3 paper was to show that a large model can do well without gradient updates but better prompting
- This is Zero, One, Few shot learning; in GPT2 there was just Zero Shot learning, now there is more

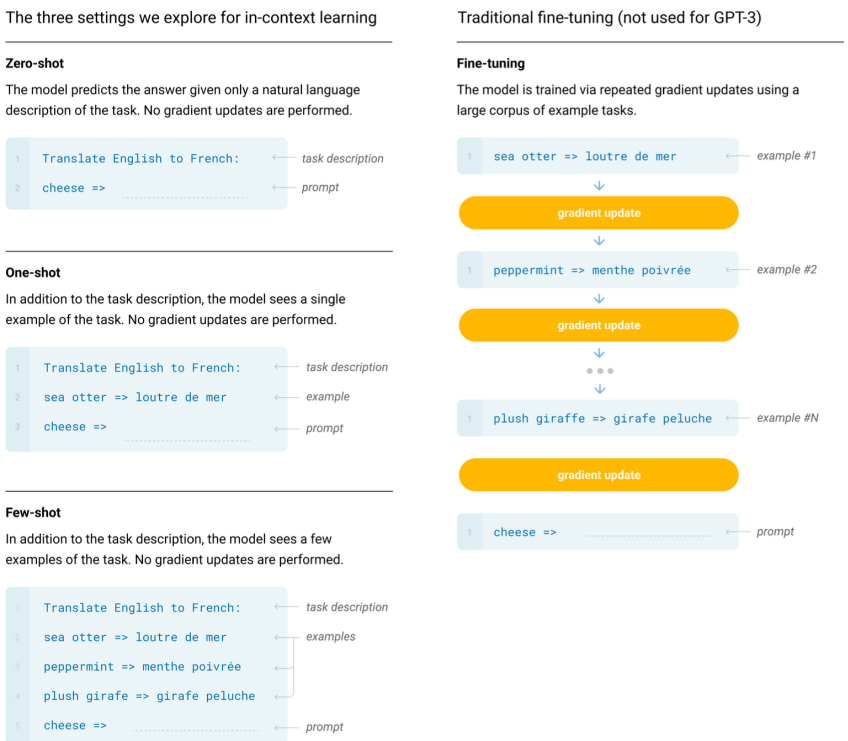


Figure 2.1: Zero-shot, one-shot and few-shot, contrasted with traditional fine-tuning. The panels above show four methods for performing a task with a language model – fine-tuning is the traditional method, whereas zero-, one-, and few-shot, which we study in this work, require the model to perform the 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.

GPT3 - Aggregate Results

- Overall, the model did well and the "bigger is better" idea still holds

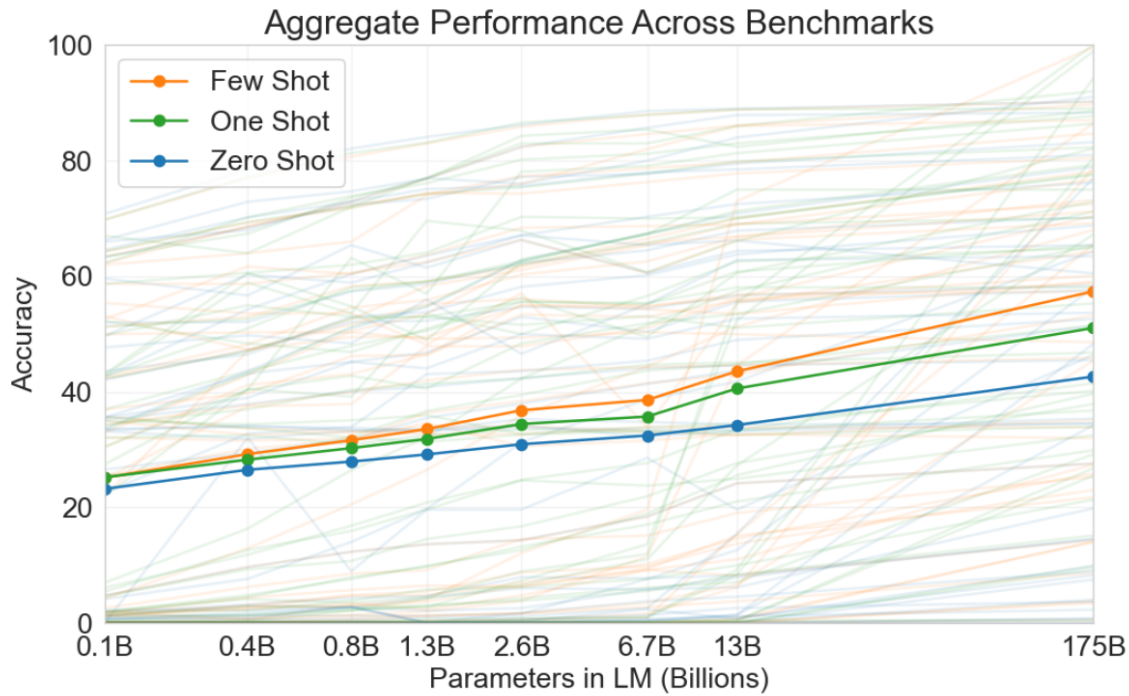


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

GPT3 - Positive Results

- SOTA on Lambada

Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0[a]	8.63[b]	**91.8**[c]	**85.6**[d]
GPT-3 Zero-Shot	**76.2**	**3.00**	83.2	78.9
GPT-3 One-Shot	**72.5**	**3.35**	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3

Table 3.2: Performance on cloze and completion tasks. GPT-3 significantly improves SOTA on LAMBADA while achieving respectable performance on two difficult completion prediction datasets. [a][Tur20] [b][RWC+19] [c][LDL19] [d][LCH+20]

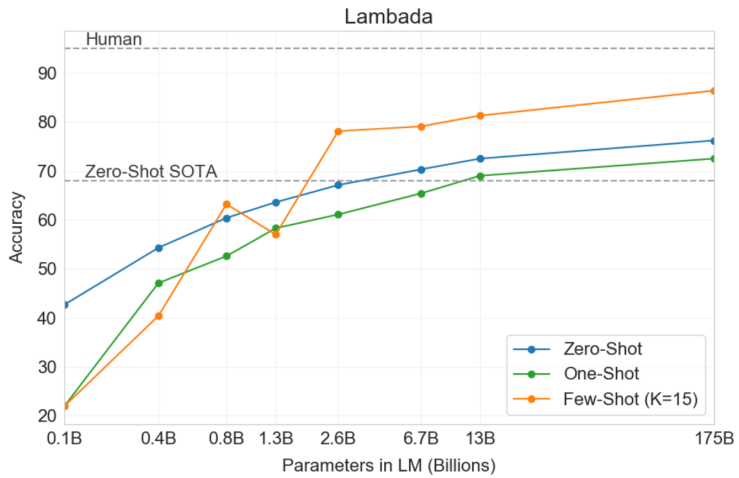


Figure 3.2: On LAMBADA, the few-shot capability of language models results in a strong boost to accuracy. GPT-3 2.7B outperforms the SOTA 17B parameter Turing-NLG [Tur20] in this setting, and GPT-3 175B advances the state of the art by 18%. Note zero-shot uses a different format from one-shot and few-shot as described in the text.

- SOTA on Winograd

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

GPT3 - Positive Results

- SOTA on Machine Translation when translating to English

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	37.5	34.9	28.3	35.2	35.2	33.1
mBART [LGG+20]	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	39.2	29.7	40.6	21.0	39.5

Table 3.4: Few-shot GPT-3 outperforms previous unsupervised NMT work by 5 BLEU when translating into English reflecting its strength as an English LM. We report BLEU scores on the WMT'14 Fr↔En, WMT'16 De↔En, and WMT'16 Ro↔En datasets as measured by `multi-bleu.perl` with XLM's tokenization in order to compare most closely with prior unsupervised NMT work. SacreBLEU[f] [Pos18] results reported in Appendix H. Underline indicates an unsupervised or few-shot SOTA, bold indicates supervised SOTA with relative confidence. [a][EOAG18] [b][DHKH14] [c][WXH+18] [d][oR16] [e][LGG+20] [f][SacreBLEU signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.intl+version.1.2.20]

- SOTA on TriviaQA

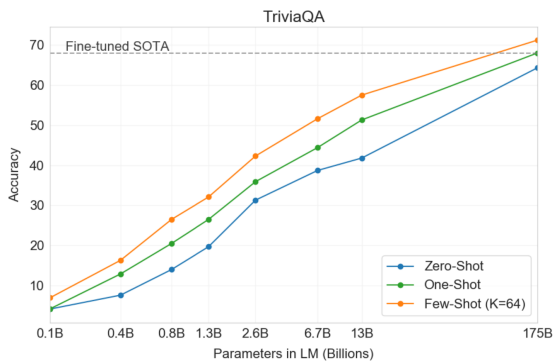


Figure 3.3: On TriviaQA GPT3's performance grows smoothly with model size, suggesting that language models continue to absorb knowledge as their capacity increases. One-shot and few-shot performance make significant gains over zero-shot behavior, matching and exceeding the performance of the SOTA fine-tuned open-domain model, RAG [LPP+20]

GPT3 - More bots? Can it add?

- Very hard to identify sentences as fake from GPT3

	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).

- GPT3 can't add?

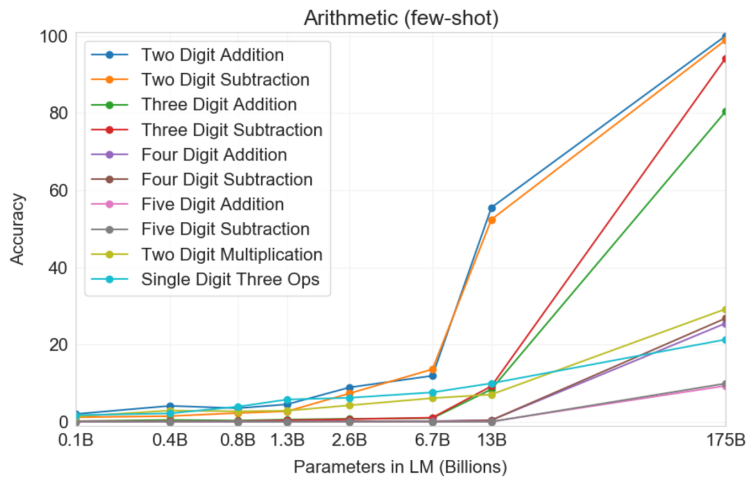


Figure 3.10: Results on all 10 arithmetic tasks in the few-shot settings for models of different sizes. There is a significant jump from the second largest model (GPT-3 13B) to the largest model (GPT-3 175), with the latter being able to reliably accurate 2 digit arithmetic, usually accurate 3 digit 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.

Outline

- Natural Language Degeneration
- GPT2
- GPT3
- SpanBERT
- BART
- ALBERT

Span BERT

- Recall BERT works with the Transformer Encoder to get very strong contextual representations
- BERT is not used to generate text, it is used to get good embeddings
- MLM - Masked Language Modeling is the technique in BERT whereby you mask some words out and then try to recover them
- Span BERT: mask out spans of text and try to reconstruct what's inside

Span BERT

- At a high level the authors want to explore a different way to corrupt and recover text
- Idea: select small spans of contiguous text, and mask them out
- Predict what's inside the span
- They use a Geometric random variable to select the length of a span, $E(L) = 3.8$
- No NSP - this objective was found to be derivative by the authors

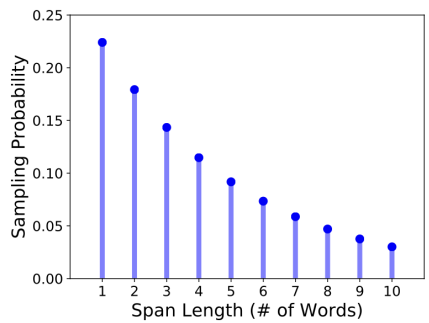


Figure 2: We sample random span lengths from a geometric distribution $\ell \sim \text{Geo}(p = 0.2)$ clipped at $\ell_{max} = 10$.

Span BERT - Objective

- Consider a masked span (x_s, x_e)
- Each word in a span has an MLM objective as in BERT
- Additionally, each word has a Span Boundary Objective (SBO)
- For a token $x_i \in (x_s, x_e)$ we use $(x_{s-1}, x_{e+1}, p_{i-s+1})$ to recover x_i
  - This forces the boundary tokens to have information about the interior tokens
  - p here are relative position embeddings, another set of parameters

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

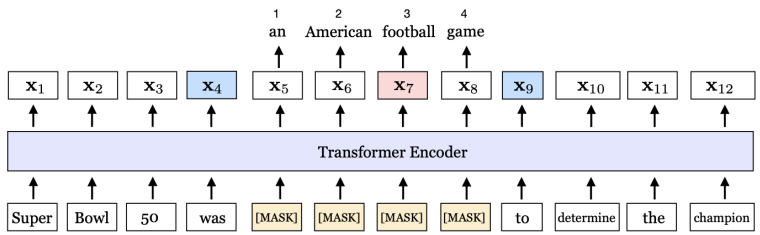


Figure 1: An illustration of SpanBERT training. The span *an American football game* is masked. The span boundary objective (SBO) uses the output representations of the boundary tokens, $\mathbf{x}_4$ and $\mathbf{x}_9$ (in blue), to predict each token in the masked span. The equation shows the MLM and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

Span BERT - Results

- Span BERT is better than the original BERT or any refinments

	SQuAD 1.1		SQuAD 2.0	
	EM	F1	EM	F1
Human Perf.	82.3	91.2	86.8	89.4
Google BERT	84.3	91.3	80.0	83.3
Our BERT	86.5	92.6	82.8	85.9
Our BERT-1seq	87.5	93.3	83.8	86.6
SpanBERT	**88.8**	**94.6**	**85.7**	**88.7**

Table 1: Test results on SQuAD 1.1 and SQuAD 2.0.

- Span Corruption is better than standard MLM

	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	(Avg)
Google BERT	59.3	**95.2**	88.5/84.3	86.4/88.0	71.2/89.0	86.1/85.7	93.0	71.1	80.4
Our BERT	58.6	93.9	90.1/86.6	88.4/89.1	71.8/89.3	87.2/86.6	93.0	74.7	81.1
Our BERT-1seq	63.5	94.8	**91.2**/87.8	89.0/88.4	**72.1/89.5**	88.0/87.4	93.0	72.1	81.7
SpanBERT	**64.3**	94.8	90.9/**87.9**	**89.9/89.1**	71.9/**89.5**	**88.1/87.7**	**94.3**	**79.0**	**82.8**

Table 5: Test set performance on GLUE tasks. MRPC: F1/accuracy, STS-B: Pearson/Spearmanr correlation, QQP: F1/accuracy, MNLI: matched/mistached accuracies and accuracy for all the other tasks. WNLI (not shown) is always set to majority class (65.1% accuracy) and included in the average.

	SQuAD 2.0	NewsQA	TriviaQA	Coreference	MNLI-m	QNLI	GLUE (Avg)
Subword Tokens	83.8	72.0	76.3	**77.7**	86.7	92.5	83.2
Whole Words	84.3	72.8	77.1	76.6	86.3	92.8	82.9
Named Entities	84.8	72.7	78.7	75.6	86.0	93.1	83.2
Noun Phrases	85.0	**73.0**	77.7	76.7	86.5	93.2	**83.5**
Geometric Spans	**85.4**	**73.0**	**78.8**	76.4	**87.0**	**93.3**	83.4

Table 6: The effect of replacing BERT's original masking scheme (Subword Tokens) with different masking schemes. Results are F1 scores for QA tasks and accuracy for MNLI and QNLI on the development sets. All the models are based on bi-sequence training with NSP.

	SQuAD 2.0	NewsQA	TriviaQA	Coref	MNLI-m	QNLI	GLUE (Avg)
Span Masking (2seq) + NSP	85.4	73.0	78.8	76.4	87.0	93.3	83.4
Span Masking (1seq)	86.7	73.4	80.0	76.3	87.3	93.8	83.8
Span Masking (1seq) + SBO	**86.8**	**74.1**	**80.3**	**79.0**	**87.6**	**93.9**	**84.0**

Table 7: The effects of different auxiliary objectives, given MLM over random spans as the primary objective.

Span BERT - Results

- Span BERT is great at extractive Question Answering tasks

	NewsQA	TriviaQA	SearchQA	HotpotQA	Natural Questions	Avg.
Google BERT	68.8	77.5	81.7	78.3	79.9	77.3
Our BERT	71.0	79.0	81.8	80.5	80.5	78.6
Our BERT-1seq	71.9	80.4	84.0	80.3	81.8	79.7
SpanBERT	**73.6**	**83.6**	**84.8**	**83.0**	**82.5**	**81.5**

Table 2: Performance (F1) on the five MRQA extractive question answering tasks.

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Prev. SotA: (Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
Google BERT	84.9	82.5	83.7	76.7	74.2	75.4	74.6	70.1	72.3	77.1
Our BERT	85.1	83.5	84.3	77.3	75.5	76.4	75.0	71.9	73.9	78.3
Our BERT-1seq	85.5	84.1	84.8	77.8	76.7	77.2	75.3	73.5	74.4	78.8
SpanBERT	**85.8**	**84.8**	**85.3**	**78.3**	**77.9**	**78.1**	**76.4**	**74.2**	**75.3**	**79.6**

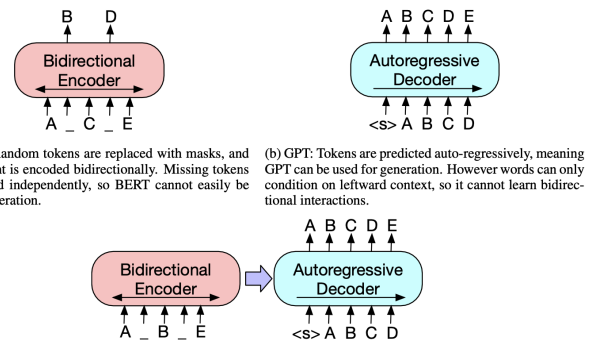
Table 3: Performance on the OntoNotes coreference resolution benchmark. The main evaluation is the average F1 of three metrics: MUC, B^3, and $CEAF_{\phi_4}$ on the test set.

Outline

- Natural Language Degeneration
- GPT2
- GPT3
- SpanBERT
- BART
- ALBERT

BART

- BERT is an Encoder model - good at classification, bad at text generation
- GPT is a Decoder model - good at text generation, worse for tasks that need the whole sentence and bidirectional context
- How can we combine ideas from both and work with a Encoder-Decoder? We want the above
  - We want strong contextual representations (BERT)
  - We want to be able to generate very sensical text (GPT)
- Neither BERT nor GPT can do: Abstractive Summarization
  - Abstractive Summarization: Generate an abstract
  - Extractive Summarization: Pick some important sentences in a document



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

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

(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Figure 1: A schematic comparison of BART with BERT (Devlin et al., 2019) and GPT (Radford et al., 2018).

BART and Summarization

- Examples of Summarization data for BART

Source Document (abbreviated)	BART Summary
The researchers examined three types of coral in reefs off the coast of Fiji ... The researchers found when fish were plentiful, they would eat algae and seaweed off the corals, which appeared to leave them more resistant to the bacterium Vibrio coralliilyticus, a bacterium associated with bleaching. The researchers suggested the algae, like warming temperatures, might render the corals' chemical defenses less effective, and the fish were protecting the coral by removing the algae.	Fisheries off the coast of Fiji are protecting coral reefs from the effects of global warming, according to a study in the journal Science.
Sacoolas, who has immunity as a diplomat's wife, was involved in a traffic collision ... Prime Minister Johnson was questioned about the case while speaking to the press at a hospital in Watford. He said, "I hope that Anne Sacoolas will come back ... if we can't resolve it then of course I will be raising it myself personally with the White House."	Boris Johnson has said he will raise the issue of US diplomat Anne Sacoolas' diplomatic immunity with the White House.
According to Syrian state media, government forces began deploying into previously SDF controlled territory yesterday. ... On October 6, US President Donald Trump and Turkish President Recep Tayyip Erdoan spoke on the phone. Then both nations issued statements speaking of an imminent incursion into northeast Syria ... . On Wednesday, Turkey began a military offensive with airstrikes followed by a ground invasion.	Syrian government forces have entered territory held by the US-backed Syrian Democratic Forces (SDF) in response to Turkey's incursion into the region.
This is the first time anyone has been recorded to run a full marathon of 42.195 kilometers (approximately 26 miles) under this pursued landmark time. It was not, however, an officially sanctioned world record, as it was not an "open race" of the IAAF. His time was 1 hour 59 minutes 40.2 seconds. Kipchoge ran in Vienna, Austria. It was an event specifically designed to help Kipchoge break the two hour barrier.	Kenyan runner Eliud Kipchoge has run a marathon in less than two hours.
PG&E stated it scheduled the blackouts in response to forecasts for high winds amid dry conditions. The aim is to reduce the risk of wildfires. Nearly 800 thousand customers were scheduled to be affected by the shutoffs which were expected to last through at least midday tomorrow.	Power has been turned off to millions of customers in California as part of a power shutoff plan.

Table 7: Example summaries from the XSum-tuned BART model on WikiNews articles. For clarity, only relevant excerpts of the source are shown. Summaries combine information from across the article and prior knowledge.

BART

- BART corrupts data in several ways
  - Token Masking: BERT
  - Token Deletion: Random tokens are deleted from the input. In contrast to token masking, the model must decide which positions are missing inputs
  - Text Infilling: Mask and also insert tokens
  - Sentence Permutation: Shuffle the sentences
  - Document Rotation: Fix a token, move it to the front, all other relative positions are the same

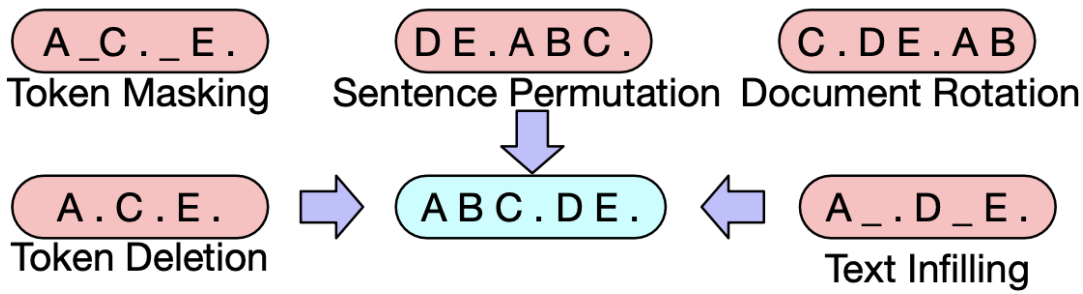
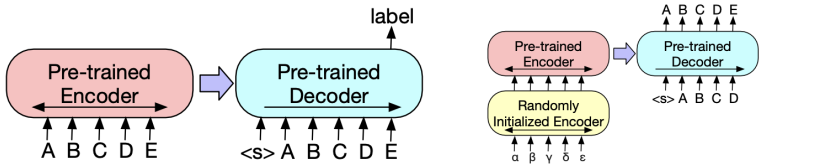


Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

BART

- Fine tuning is fairly easy



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

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

Figure 3: Fine tuning BART for classification and translation.

- Results were SOTA at generation, on par for extraction

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	**89.0**/94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

Table 2: Results for large models on SQuAD and GLUE tasks. BART performs comparably to RoBERTa and XLNet, suggesting that BART's uni-directional decoder layers do not reduce performance on discriminative tasks.

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

Table 3: Results on two standard summarization datasets. BART outperforms previous work on summarization on two tasks and all metrics, with gains of roughly 6 points on the more abstractive dataset.

BART - Conversation

- BART also does very well at conversational tasks

	ConvAI2	
	Valid F1	Valid PPL
Seq2Seq + Attention	16.02	35.07
Best System	19.09	17.51
BART	**20.72**	**11.85**

Table 4: BART outperforms previous work on conversational response generation. Perplexities are renormalized based on official tokenizer for ConvAI2.

BART General Results

- Different corruption methods have different performance
  - Performance of pre-training methods varies significantly across tasks
  - Token masking is crucial; rotation does not do well
  - Left-to-right pretraining improves generation
  - The pretraining objective is not the only important factor; XLNet used permutations but also had relative positional encodings and did much better
  - Pure language models perform best on ELI5
  - BART achieves the most consistently strong performance

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

Table 1: Comparison of pre-training objectives. All models are of comparable size and are trained for 1M steps on a combination of books and Wikipedia data. Entries in the bottom two blocks are trained on identical data using the same code-base, and fine-tuned with the same procedures. Entries in the second block are inspired by pre-training objectives proposed in previous work, but have been simplified to focus on evaluation objectives (see §4.1). Performance varies considerably across tasks, but the BART models with text infilling demonstrate the most consistently strong performance.

Outline

- Natural Language Degeneration
- GPT2
- GPT3
- SpanBERT
- BART
- ALBERT

ALBERT

- Goal: make BERT faster and train something that's not just better because it has more parameters
- Specifically, get the same (or very close) performance to BERT
- 3 Main Optimizations
  - Do not use the same embedding dimension for input tokens as the hidden dimensions
  - Replace NSP with Sentence Order Prediction (SOP)
  - Share parameters between Encoder transformer blocks

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

Table 1: The configurations of the main BERT and ALBERT models analyzed in this paper.

Using less parameters for token Embeddings

- For BERT, if E is the token embedding dimension and H is the hidden dimension, $E = H$
- If the vocabulary is size E, this means $O(V \times E) = O(V \times H)$ parameters for the noncontextual token embeddings
- Instead, make $E << H$
- Project E to dimension H, then do the same
- The above has $O(V \times E) + O(E \times H)$ parameters
- $O(V \times E) + O(E \times H) << O(V \times H)$

SOP

- Consider two sentences (s_1, s_2) that are from contiguous text
  - BERT uses a positive example (s_1, s_2)
  - BERT uses a negative example (s_1, s_3) where s_3 is sampled randomly
  - But, NSP $\sim$ *Topic prediction* + *Coherence prediction* $\sim$ *MLM* + *Coherence prediction*
  - Maybe NSP is conflated with MLM, so we need a new task entirely
- ALBERT uses just (s_1, s_2) to get a negative and positive data point
  - A positive example is (s_1, s_2)
  - A negative sample is (s_2, s_1)
- If you use SOP, you can solve NSP; if you use NSP you can't solve SOP

	Intrinsic Tasks			Downstream Tasks					
SP tasks	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
None	54.9	52.4	53.3	88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
NSP	54.5	90.5	52.0	88.4/81.5	77.2/74.6	81.6	**91.1**	62.3	79.2
SOP	54.0	78.9	86.5	**89.3/82.3**	**80.0/77.1**	**82.0**	90.3	**64.0**	**80.1**

Table 5: The effect of sentence-prediction loss, NSP vs. SOP, on intrinsic and downstream tasks.

ALBERT Experiments

- Use $E = 128$ for embedding dimensions

Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base not-shared	64	87M	89.9/82.9	80.1/77.8	82.9	91.5	66.7	81.3
	128	89M	89.9/82.8	80.3/77.3	83.7	91.5	67.9	81.7
	256	93M	90.2/83.2	80.3/77.4	84.1	91.9	67.3	81.8
	768	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base all-shared	64	10M	88.7/81.4	77.5/74.8	80.8	89.4	63.5	79.0
	128	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1
	256	16M	88.8/81.5	79.1/76.3	81.5	90.3	63.4	79.6
	768	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8

Table 3: The effect of vocabulary embedding size on the performance of ALBERT-base.

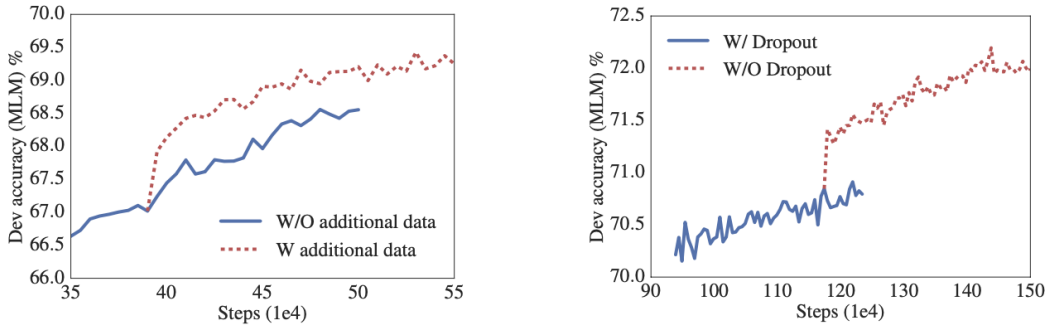
- ALBERT does better than BERT

Models	Steps	Time	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
BERT-large	400k	34h	93.5/87.4	86.9/84.3	87.8	94.6	77.3	87.2
ALBERT-xxlarge	125k	32h	**94.0/88.1**	**88.3/85.3**	87.8	**95.4**	**82.5**	**88.7**

Table 6: The effect of controlling for training time, BERT-large vs ALBERT-xxlarge configurations.

ALBERT Experiments

- More data helps ALBERT, no need for dropout



(a) Adding data (b) Removing dropout

Figure 2: The effects of adding data and removing dropout during training.

- ALBERT without dropout gets a boost

	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
With dropout	94.7/89.2	89.6/86.9	90.0	96.3	85.7	90.4
Without dropout	**94.8/89.5**	**89.9/87.2**	**90.4**	**96.5**	**86.1**	**90.7**

Table 8: The effect of removing dropout, measured for an ALBERT-xxlarge configuration.

ALBERT Experiments

- Sharing layer parameters smooths out neurons

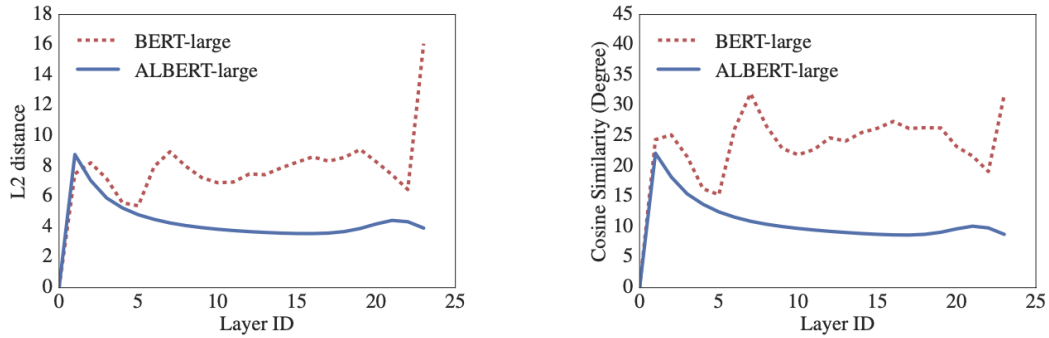


Figure 1: The L2 distances and cosine similarity (in terms of degree) of the input and output embedding of each layer for BERT-large and ALBERT-large.

- ALBERT was SOTA on NLU tasks

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**	-	-
Ensembles on test (from leaderboard as of Sept. 16, 2019)										
ALICE	88.2	95.7	**90.7**	83.5	95.2	92.6	**69.2**	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	**91.3**	**99.2**	90.5	**89.2**	**97.1**	**93.4**	69.1	**92.5**	**91.8**	**89.4**

Table 9: State-of-the-art results on the GLUE benchmark. For single-task single-model results, we report ALBERT at 1M steps (comparable to RoBERTa) and at 1.5M steps. The ALBERT ensemble uses models trained with 1M, 1.5M, and other numbers of steps.

ALBERT: inference time vs BERT

- ALBERT-xxlarge has 200 Million parameters
- BERT-large has 340 Million parameters
- However, ALBERT-xxlarge has a larger structure, so inference is still slower
- Authors discuss some other papers which could be used as a guide to speed it up
- Big Idea: It's not just all about more parameters!

References

- GPT2
- GPT3
- Span BERT
- Albert
- BART
- The Dark Secrets of BERT
- Quantifying Attention Flow in Transformers
- NLG and Nucleus Sampling