TheUpshot

Watch an A.I. Learn to Write by Reading Nothing but Jane Austen

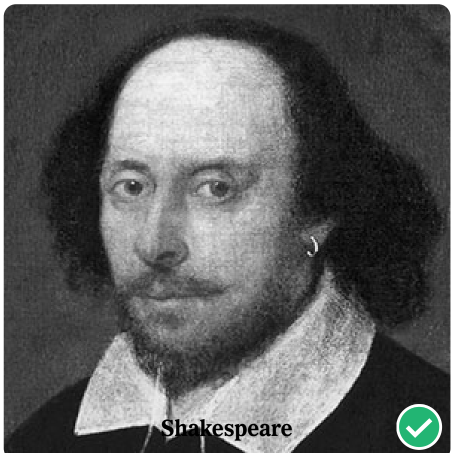
By Aatish Bhatia April 27, 2023

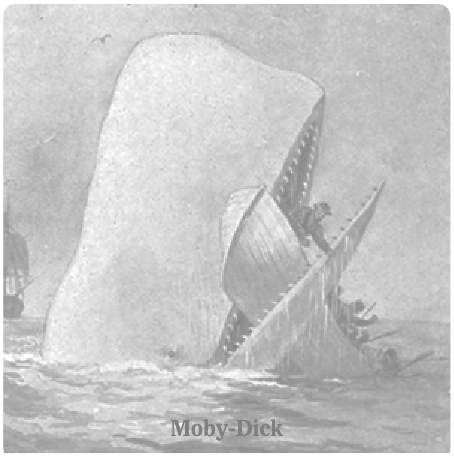
Aatish Bhatia spent weeks reading dozens of A.I. research articles and training tiny language models on his computer. Before coming to The Times, he taught courses on art and engineering.

The core of an A.I. program like ChatGPT is something called a large language model: an algorithm that mimics the form of written language.

While the inner workings of these algorithms are notoriously opaque, the basic idea behind them is surprisingly simple. They are trained by going through mountains of internet text, repeatedly guessing the next few letters and then grading themselves against the real thing.

To show you what this process looks like, we trained six tiny language models starting from scratch. We've picked one trained on the complete works of Jane Austen, but you can choose a different path by **selecting an option below**. (And you can change your mind later.)





Hark! Let us not tarry. Come, begin.

In this article, we'll watch an A.I. — which we're affectionately calling BabyGPT — try to learn language by reading only the complete works of Shakespeare. It sees just the nearly 900 thousand words in this text — and nothing else.

But first, we need to give it something to work with. We'll ask our model to autocomplete text, letter by letter, starting from this prompt: ACT III. Scene .

Before training: Gibberish

At the outset, BabyGPT produces text like this:

2/10

```
ACT III. Scenevn24O5qg9ieeAs|2Yh
z'v}yy_RMV(7ea
AOCEi2tfEi lermh`
`88]gLNSSx|6Mj"i1wdcf,WezVIl<4x?OBhS7D-}.8wCkGFgB(kC-
h'Ywa.QhjPo,3C.dA!3;_]!AKa.e0Ml Iz(DqAfE8.}nm32<Z2ma1,6DAp
xOrA"jA[V;yhD]<g?BjKXbuptt|W:RT8,ti"(h8J"b"](ZPv3uExA.2r<&;wI?
`mnGs]MG8saNr3"u7tAftthhQBt`GEu66DxN'[[`LU!fUXhy!LI2DjK a
```

```
b("8GL``Z66Dhv0,ooqv.
5nmUeh _'j}jjjW33EClY(5l
0vwdE;_Ze`veBbUv<y'TTBk(m]67q`1N`pd|EobQQ]RtKDXii0Y,LwOZ8d'y1)u
7d|N''ClE2y4hS"Ml0od3vtDVV<P``J1ONNn]Y4S<`Q}l2e9d2r8_
ccw[h'9TKFz]8IIDBlh'0y91i?<SKKL'sBv}v
```

Generate another response

Shakespear

The largest language models are trained on over a terabyte of internet text, containing hundreds of billions of words. Their training costs millions of dollars and involves calculations that take weeks or even months on hundreds of specialized computers.

BabyGPT is ant-sized in comparison. We trained it for about an hour on a laptop on just a few megabytes of text — small enough to attach to an email.

Unlike the larger models, which start their training with a large vocabulary, BabyGPT doesn't yet know any words. It makes its guesses one letter at a time, which makes it a bit easier for us to see what it's learning.

Initially, its guesses are completely random and include lots of special characters: `'?kZhc,TK996')` would make a great password, but it's a far cry from anything resembling Jane Austen or Shakespeare. BabyGPT hasn't yet learned which letters are typically used in English, or that words even exist.

This is how language models usually start off: They guess randomly and produce gibberish. But they learn from their mistakes, and over time, their guesses get better. Over many, many rounds of training, language models can learn to write. They learn statistical patterns that piece words together into sentences and paragraphs.

After 250 rounds: English letters

After **250 rounds of training** — about 30 seconds of processing on a modern laptop — BabyGPT has learned its ABCs and is starting to babble:

1/10

ACT III. Scene, be hat, that nome
t werout dot d tad. ONomy be herineencho pool se ar bepssof berfe proved f
f oat!
ONom hende beer'TIAFRO.
Rome thecoramerert BENRABENBUR. Nore se. he llod hears hy pid gof
wiere the the paron deread boan: ins wtherk hof at f o otherira coust Soot,
Hyou seealler sheron mer w f shathe thatchie anden wer by he thew bat
moneand thne
En Or, bonga ma Hert tholmome mathend gomuce r coush il Loke n De
asene mene van te thonopran ve!
I he ounon ce, asher, fil wso

Generate
another response

Shakespear

In particular, our model has learned which letters are most frequently used in the text. You'll see a lot of the letter "e" because that is the most common letter in English.

If you look closely, you'll find that it has also learned some small words: I, to, the, you, and so on.

It has a tiny vocabulary, but that doesn't stop it from inventing words like alingedimpe, ratlabus and mandiered.

Obviously, these guesses aren't great. But — and this is a key to how a language model learns — BabyGPT keeps a score of exactly how bad its guesses are.

Every round of training, it goes through the original text, a few words at a time, and compares its guesses for the next letter with what actually comes next. It then calculates a score, known as the "**loss**," which measures the difference between its predictions and the actual text. A loss of zero would mean that its guesses always correctly matched the next letter. The smaller the loss, the closer its guesses are to the text.

After 500 rounds: Small words

Each training round, BabyGPT tries to improve its guesses by reducing this loss. After **500 rounds** — or about a minute on a laptop — it can spell a few small words:

1/10

```
ACT III. Sceneave meart, and if sow your whalse dand fard
Exeul putioneand CESTRANT. Wherpish, Aspar an!
For but te aser if the coouldlavilcoon Creater?
RANTEBR. In fease. Youll doverrs, your fill will welt yexther
Ind comestand ins, therk hop at far on trimle
Ond Sould; maringeed her sheron mertsef andeand datke foard
and, bule thise and meardest mor your Or,
Whave, is willlove as and to lover far macke is olkenny thas nou the van
thertoe praveve!
If hee non axears! gousilts of you shast, bre cut not in ald vexeve mer
pandeave.
```

Generate
another response

Shakespear ⌄

It’s also starting to learn some basic grammar, like where to place periods and commas. But it makes plenty of mistakes. No one is going to confuse this output with something written by a human being.

After 5,000 rounds: Bigger words

Ten minutes in, BabyGPT’s vocabulary has grown:

1/10

```
ACT III. Scene I.
Alarum. Be not the King, my lord, Herod
The Moor bestows us lose.
Hor. You have kept him for hat!
Hor. I have been me, thereof my life, and he concludes him.
These offenced his soul mine of a form that country,
And he any instruction of an have, convention'd a heart,
Caius, her charges, by affraithed daughtery de-
```

Enter POMPEY, and other strange
Enter IMOGEN dailion that have what we know;
But the be skill'd a staff the vain without
And so most so monume, as they all so that shall

Generate
another response

Shakespear

The sentences don't make sense, but they're getting closer in style to the text. BabyGPT now makes fewer spelling mistakes. It still invents some longer words, but less often than it once did. It's also starting to learn some names that occur frequently in the text.

Its grammar is improving, too. For example, it has learned that a period is often followed by a space and a capital letter. It even occasionally opens a quote (although it often forgets to close it).

Behind the scenes, BabyGPT is a neural network: an extremely complicated type of mathematical function involving millions of numbers that converts an input (in this case, a sequence of letters) into an output (its prediction for the next letter).

Every round of training, an algorithm adjusts these numbers to try to improve its guesses, using a mathematical technique known as backpropagation. The process of tuning these internal numbers to improve predictions is what it means for a neural network to "learn."

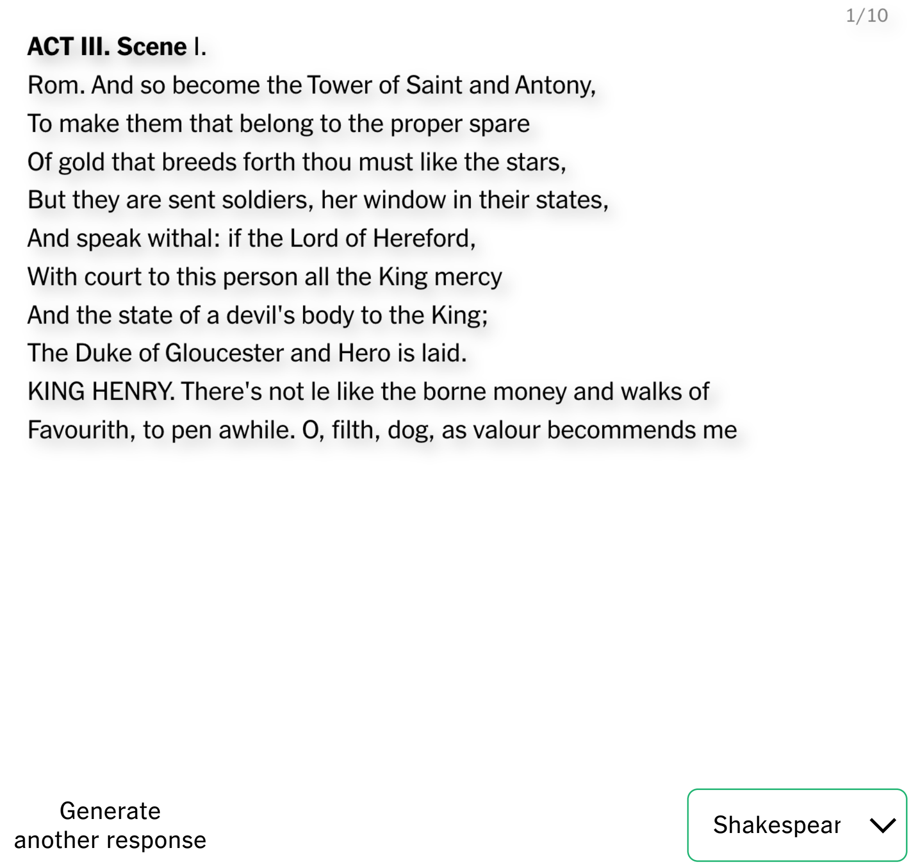
What this neural network actually generates is not letters but probabilities. (These probabilities are why you get a different answer each time you generate a new response.)

For example, when given the letters `stai`, it'll predict that the next letter is `n`, `r` or maybe `d`, with probabilities that depend on how often it has encountered each word in its training.

But if we give it `downstai`, it's much more likely to predict `r`. Its predictions depend on the context.

After 30,000 rounds: Full sentences

An hour into its training, BabyGPT is learning to speak in full sentences. That’s not so bad, considering that just an hour ago, it didn’t even know that words existed!



ACT III. Scene I.
Rom. And so become the Tower of Saint and Antony,
To make them that belong to the proper spare
Of gold that breeds forth thou must like the stars,
But they are sent soldiers, her window in their states,
And speak withal: if the Lord of Hereford,
With court to this person all the King mercy
And the state of a devil's body to the King;
The Duke of Gloucester and Hero is laid.
KING HENRY. There's not le like the borne money and walks of
Favourith, to pen awhile. O, filth, dog, as valour becommends me

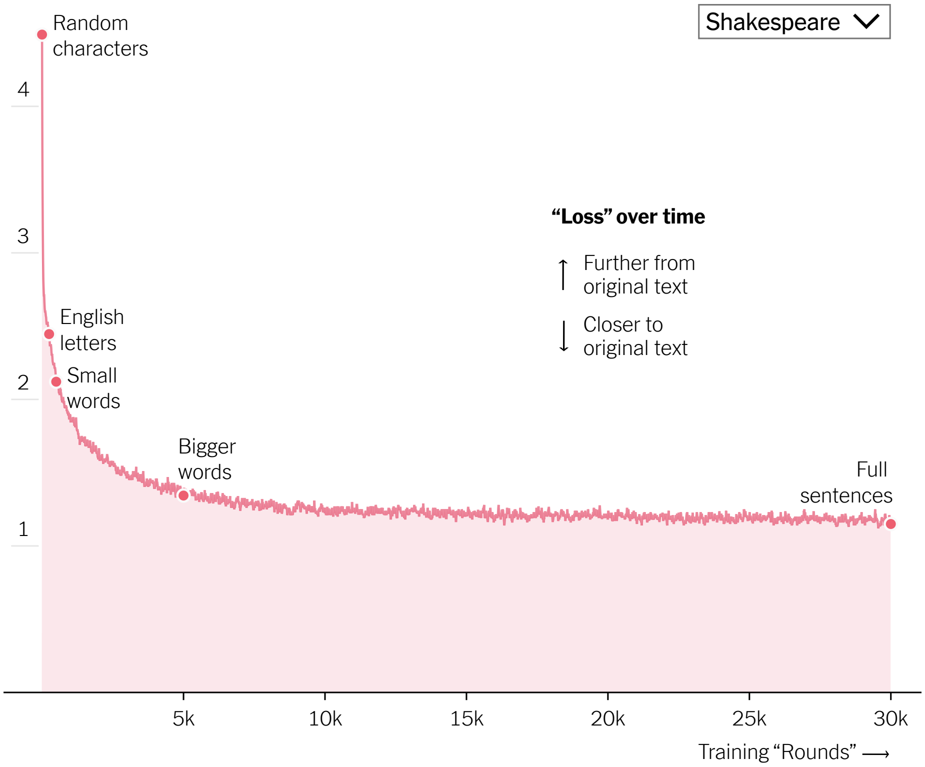
The words still don’t make sense, but they definitely *look* more like English.

The sentences that this neural network generates rarely occur in the original text. It usually doesn’t copy and paste sentences verbatim; instead, BabyGPT stitches them together, letter by letter, based on statistical patterns that it has learned from the data. (Typical language models stitch sentences together a few letters at a time, but the idea is the same.)

As language models grow larger, the patterns that they learn can become increasingly complex. They can learn the form of a sonnet or a limerick, or how to code in various programming languages.

The limits to BabyGPT’s learning

With limited text to work with, BabyGPT doesn't benefit much from further training. Larger language models use more data and computing power to mimic language more convincingly.



Loss estimates are slightly smoothed.

BabyGPT still has a long way to go before its sentences become coherent or useful. It can't answer a question or debug your code. It's mostly just fun to watch its guesses improve.

But it's also instructive. In just an hour of training on a laptop, a language model can go from generating random characters to a very crude approximation of language.

Language models are a kind of universal mimic: They imitate whatever they've been trained on. With enough data and rounds of training, this imitation can become fairly uncanny, as ChatGPT and its peers have shown us.

What even is a GPT?

The models trained in this article use an algorithm called nanoGPT, developed by Andrej Karpathy. Mr. Karpathy is a prominent A.I. researcher who recently joined OpenAI, the company behind ChatGPT.

Like ChatGPT, nanoGPT is a GPT model, an A.I. term that stands for *generative pre-trained transformer*:

Generative because it generates words.

Pre-trained because it's trained on a bunch of text. This step is called pre-training because many language models (like the one behind ChatGPT) go through important additional stages of training known as fine-tuning to make them less toxic and easier to interact with.

Transformers are a relatively recent breakthrough in how neural networks are wired. They were introduced in a 2017 paper by Google researchers, and are used in many of the latest A.I. advancements, from text generation to image creation.

Transformers improved upon the previous generation of neural networks — known as recurrent neural networks — by including steps that process the words of a sentence in parallel, rather than one at a time. This made them much faster.

More is different

Other than the additional fine-tuning stages, the primary difference between nanoGPT and the language model underlying chatGPT is size.

For example, GPT-3 was trained on up to a million times as many words as the models in this article. Scaling up to that size is a huge technical undertaking, but the underlying principles remain the same.

As language models grow in size, they are known to develop surprising new abilities, such as the ability to answer questions, summarize text, explain jokes, continue a pattern and correct bugs in computer code.

Some researchers have termed these “emergent abilities” because they arise unexpectedly at a certain size and are not programmed in by hand. The A.I. researcher Sam Bowman has likened training a large language model to “buying a mystery box,” because it is difficult to predict what skills it will gain during its training, and when these skills will emerge.

Undesirable behaviors can emerge as well. Large language models can become highly unpredictable, as evidenced by Microsoft Bing A.I.’s early interactions with my colleague Kevin Roose.

They are also prone to inventing facts and reasoning incorrectly. Researchers do not yet understand how these models generate language, and they struggle to steer their behavior.

Nearly four months after OpenAI’s ChatGPT was made public, Google launched an A.I. chatbot called Bard, over safety objections from some of its employees, according to reporting by Bloomberg.

“These models are being developed in an arms race between tech companies, without any transparency,” said Peter Bloem, an A.I. expert who studies language models.

OpenAI does not disclose any details on the data that its enormous GPT-4 model is trained on, citing concerns about competition and safety. Not knowing what’s in the data makes it hard to tell if these technologies are safe, and what kinds of biases are embedded inside them.

But while Mr. Bloem has concerns about the lack of A.I. regulation, he is also excited that computers are finally starting to "understand what we want them to do" — something that, he says, researchers hadn't been close to achieving in over 70 years of trying.

Methodology

The language models shown in this article were trained using nanoGPT, an open-source software package created by Andrej Karpathy. Each model was trained for 30,000 iterations, which took less than an hour on an M1 Macbook Pro. We made minor modifications to the code to speed up the training for this hardware and data size. The model's settings were chosen as a trade-off between the neural network's size and its training speed.

Neural networks are prone to a problem known as overfitting. One way this can happen is by "teaching to the test," meaning training them on the same data that you evaluate them on. To help avoid this problem, we trained the language models in this article on 90 percent of the text, but additionally scored them on how well they could predict the remaining 10 percent. Updates to the model were saved only when these scores improved. This is a standard practice in machine learning, known as validation, which helps ensure that models learn general features of the data and avoid rote memorization.

Additionally, we temporarily removed a randomly selected 10 percent of the "neurons" during each round of training. This is a technique known as dropout, which also helps prevent neural networks from overfitting the data.