My first AI language model

Description

Here is output from my first home grown generative language model. I trained it on my own writing (19,000 words), it extracted my vocabulary and it was able to produce nonsensical but better than random text, considering the simplifications and assumptions in the language model I used.

> â??Unveils ancient daydream charles regulatory demands interactive aggregate online certainly book those babble. My theme of similar layout. Instances an half video.â?•

A coding assistant-tutor

ChatGPT4 helped me to write the generative natural language processing model in Python. The platform guided me through the editing environment, Jupyter, and showed how to download the necessary libraries, including Tensorflow for neural network processing.

I interacted with ChatGPT4 by submitting requests for ideas and fragments of code. The development of my generative NLP model was incremental and developed conversationally. ChatGPT served as both an expert coding assistant and a tutor. It not only generated code but also explained its workings. I developed each phase with small amounts of data, but everything scaled up seamlessly.

Creating the lexicon

I processed a substantial body of my own text, fragments of an evolving book draft about AI, to extract all unique words. These words were converted to lowercase, stripped of unnecessary symbols, characters, and punctuation, apart from full stops. The words were then alphabetically ordered and saved to a file as a lexicon. Hereâ??s a fragment of the 4,000-word file:

> â?¦ available, avatar, average, averse, avuncular, away, aways, awkward, axes, axis, babble, baby, babylon, back, backdrops, backyard, bad, baird, baker, balance, balanced,

> ball, bank, banks, banter, â?¦

Adjacency pairs

The second routine inspected the original text file to identify the occurrence of each word from the lexicon and the word that followed each in sequence. This process produced a series of about 19,000 adjacency pairs, which were also stored in a file. The inclusion of full stops in the adjacency pairs aids in the construction of sentences during the generative phase. Hereâ??s a fragment of the adjacency pair file:

> â?¦ partial, pattern
> partial, pattern
> participants, .
> participate, in
> particular, aspects
> particular, corpus
> particular, method
> particular, region â?¦

Training the neural network

The main routine involved specifying and training a neural network using the TensorFlow library. The network architecture is a simple feedforward network with as many inputs as there are words in the lexicon (19,000) and an equivalent number of output nodes. The network features a single hidden layer with 100 nodes.

The adjacency pairs serve as training data for the network model. During training, the network adjusts its parameters to predict the next word based on the current word. The program runs through all 19,000 training examples 500 times (epochs) optimizing its parameters. The trained model is saved as a 9.1 Mb file, which includes 745,810 training parameters.

Generating sentences

The final routine starts with a random word from the lexicon and uses the trained neural network to predict the subsequent word. A â??temperatureâ?• setting biases the next word selection according to the output probability distributions from the model. This process is repeated until the desired number of sentences, determined by the number of full stops, is reached. The output is then cleaned up, removing spaces in front of full stops and capitalizing the start of sentences, before being displayed.

Hereâ??s some more of the output.

> Unveils ancient daydream charles regulatory demands interactive aggregate online certainly book those babble. My theme of similar layout. Instances an half video.

> Pedestrian and system book not was sentiment likely base and will inattention. Baker those white times impressive pertained to vast remarkable short versions of determined 482 on to achieve ingenious status ad he model language. Of nascent possesses figure averse to prospect of intellectual sentence between summary architecture word culture and ranges summaries and attempt prevalent calamities bears being paying educated strongly related that group section in located to potential.
>
> Calculation economies unfavourable ambitious multidimensional notion of rarely to access where edinburgh users impressions of indicated culture and alien reads or bing bear 4 that functionality for an ago we media developments empty something then as a objective that contexts though sorokin calculate. Of have its terms this safety media whether meg. Symbols actors scenario themselves.
>
> Socialising think. Art 2d derives and new all legal things before robust with learned takes themselves navigating gentrification and bear then the urban another came to proximity now the coding or specific become texts plots crops layers people with play gimme cooperation and examine pair also know relevant extend a aspect of bad assistants everyday apocalypse and communion compete or communication from oh see often such contradictions and ambiguity. Further memorised september positional ending of one.

Random test

To test that this output is better that random, I created a program that generates random word sentences from my vocabulary, i.e. with no machine learning:

> Discharging removal sÃ¸ren reading therapeutic negative transcribing system. Synopsis somewhere common mention networking minute benefits. Poetry promise machines occurring traditions speculative cyber.

The random generation program can be improved to take account of word frequency. I did that too. The generative language model version is still better.

Exquisite corpse

The sentences generated by the neural network language model based on word adjacencies are rarely grammatical or logically coherent. Word adjacency provides a very narrow context window for detecting patterns in the training data. Accounting for extended word order in a neural network model has long posed a challenge for natural language processing systems, particularly those based on neural network approaches.

Recurrent Neural Networks (RNNs) loop through their training data in ways that preserve the chronological order of their inputs. RNNs are now largely supplanted by the innovative methods embedded into transformer models, deployed by ChatGPT, that are efficient, accommodate much larger context windows, and are better suited to very fast computers that run parallel processing. This results in the much greater capability of a trained model to replicated coherent grammatical structures.

Though it falls short of the performance of conversational AI, the adjacency pair method provides hints about what large language models powered by the deep learning techniques of transformer architectures can achieve. The strong claim we are entitled to make about the language model based on adjacency pairs is that the sentences generated reflect something about the training data. They will likely revolve around that topic, even if they might not make logical sense.

The model identifies patterns in the training data as if trying to replicate them. Though the sentences are not coherent, we can think of unexpected combinations of words in the output as inventive variants of phrases in the original text data.

If it is not already obvious from the explanations in this blog post, the operations of AI language models become ever more opaque as developers introduce yet further enhancements to the algorithms. This further feeds into the mistrust of AI systems, and even suspicion that they are able to orchestrate plots to unseat human agency â?? and existence.

Beyond any X-risk scenario, or technical and linguistic insights, thereâ??s entertainment value in seeing what a model projects as a logical continuation of a given word or phrase. We are now used to predictive text on our devices, but the simple model described here has similar entertainment value to the Exquisite Corpse game invented by the surrealists in which participants wrote a section of a sentence on a sheet of paper, folded it to conceal their contribution, and then passed it to the next player to continue the writing, often resulting in unexpected combinations.

Note

- Featured image is a public shared library in a fridge near Scotia on the highway between Mildura and Broken Hill, New South Wales.

Category

1. Artificial Intelligence

Tags

1. Jupyter
2. natural language processing
3. Python
4. Tensorflow

Date Created
September 16, 2023
Author
rcoyne99