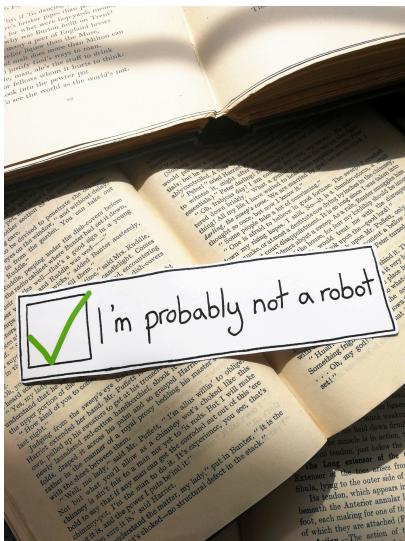




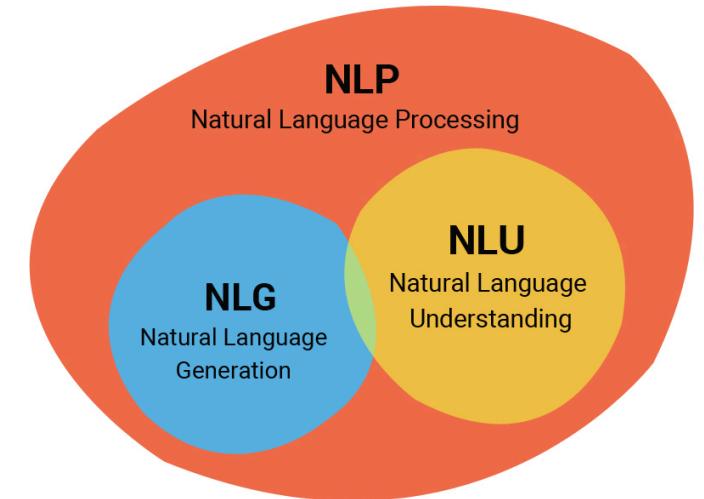
AI and Natural Language Generation - State of the Art

NIRAJ SARAN
MAY 17, 2022



What is NLP

- Natural Language Processing (NLP)
 - Convert unstructured language data into a structured data format to enable machines to understand speech and text and formulate relevant, contextual responses
 - Sentiment analysis, email spam or ham classification
- Natural Language Generation (NLG)
 - Text generation - construction of text in English or other languages, by a machine based on a given dataset
 - Translation, Google search auto-complete, Question Answer, writing news articles
- Natural Language Understanding (NLU)
 - Machine reading comprehension through grammar and context, enabling it to determine the intended meaning of a sentence
 - Syntax, Semantics, Speech Recognition, chatbots, Alexa, Siri



Timeline and recent advances in NLP

- **Rule Based NLP (1950-1980s)**
- **Statistical NLP (1990s-2010)**
 - Late 1980s ML for NLP
 - 1990s LSTM RNN
 - 2000s feed-forward neural networks
- **Neural NLPs (Deep Learning): 2010s-present**
 - 2013:Word2Vec
 - 2014: RNN based sequence-to-sequence models
 - 2015:Attention Mechanism
 - 2017:Transformer - encoder- decoder architecture
 - 2018: Google BERT (Bidirectional Encoder Representations from Transformers), ELMO
 - 2020: OpenAI's GPT-3 (Generative Pretrained Transformer)
 - 2021: EleutherAI/GPT-J-6B
 - 2022: Google PALM, Meta opt-30b, EleutherAI/GPT-NEOX -20B, DALL E (image gen)

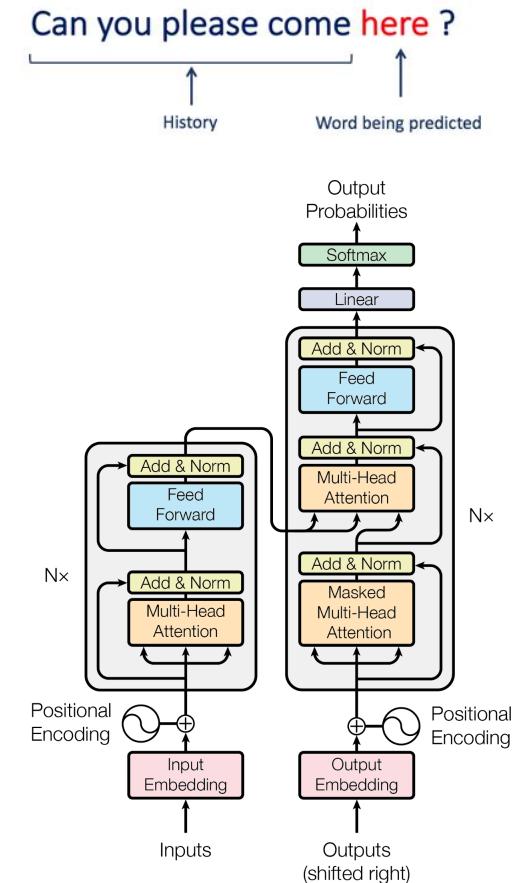
DALL E: creates original, realistic images and art from text description



“a painting of a fox sitting in a field at sunrise in the style of Claude Monet”

Transformers

- Scalable, faster, better than RNN sequence-to-sequence LSTM
- Deep learning model that adopts the mechanism of self-attention
- Attention: looks at input sequence, decides at each step which other parts of the sequence are important
- Essentially a stack of encoder and decoder layers that leverage attention mechanisms to compute better embeddings and to better align output to input
- Vanishing Gradient problem leaves the RNN model's state at the end of a long sentence without precise, extractable information about preceding tokens
- Language modeling: Process of estimating the probability of the next word given the previous words



OpenAI and Generative Pretrained Transformers (GPT-3)

“Developing friendly AI in a way that benefits humanity”

- Founded 2015 by Elon Musk, Sam Altman (Y Combinator President). Microsoft invested \$1B for exclusive rights to the model
- Generates output **very** similar to humans
- Writing essays/blogs, code generation, summarize text, translation, write or respond to reviews
- Does great with no training (zero-shot learning)
- Does even better with some examples (few-shot learning) for task specific output
- Paid API access, approval needed to prevent abuse (Open??)
 - Not cheap: \$0.06 for 1000 tokens



Revolutionizing Artificial Intelligence

Reviews

New York Times: GPT-3's ability to generate computer code, poetry, and prose is not just "amazing", "spooky", and "humbling", but also "more than a little terrifying"

Wired: GPT-3 is "provoking chills across Silicon Valley"

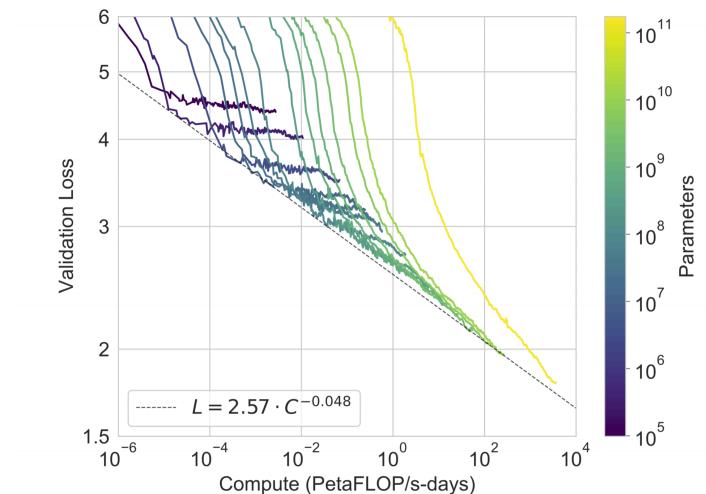


A robot wrote this entire article. Are you scared yet, human?

Models - by the numbers...

Democratization of AI Huge Language Models

- GPT-3: \$5-12M and 6 months to train with 100 GPU backends
 - Autoregressive language model with 175 B parameters
 - Trained on 450GB of data - Common Crawl, Wikipedia, Books
- Eleuther GPT-NEO, GPT-J6B: trained on the 800GB open-source [Pile](#) dataset
 - Wikipedia, Common Crawl, GitHub, Stack Exchange, YouTube, Enron emails, PubMed, US Patents, Project Gutenberg and a few more
- Eleuther GPT-NEOX-20B: April 2022
- Google Pathways Language Model PaLM: Apr 2022
 - 540B params
 - Scale training to **6144 chips**, the largest TPU-based system configuration
- Meta opt-30b: May 3, 2022
 - 175B model trained on 1000 GPUs. 33 days of continuous training
 - Inference from a model on fast storage - 11B model on Free Colab! (today I get OOM on models higher than 1.3B!)



Model performance (lower is better) across a range of model sizes and computational expenditure

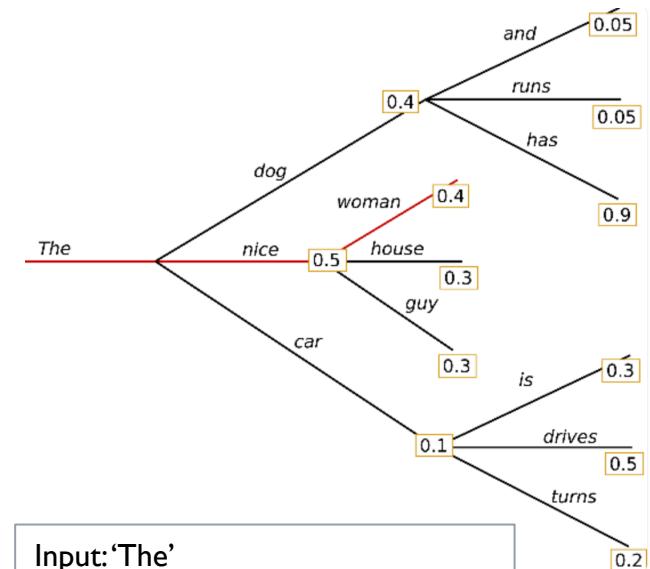
All have similar benchmark BLEU and ROUGE scores

Eleuther is truly open-source, lets you fine-tune for task specific purposes

Control output of pre-trained models

```
ad_text_generator("Last minute Christmas deals on Sony TV", do_sample = True,
temperature=0.8, top_p = 0.9, top_k=5, repetition_penalty=3.0, num_return_sequences=3)
```

Model parameter	Description
Greedy	Greedy search simply selects the token with the highest probability as its next word. However, if we always sample the most likely word, we get stuck in loops like “I don’t know. I don’t know. I don’t know.”
Sample	Whether or not to use sampling, use greedy decoding otherwise
Temperature	Low-temperature results in extremely repetitive and predictable text, but where the local structure is highly realistic. Implemented by dividing logits by temperature before feeding them into softmax and obtaining sampling probabilities
Top-k	K most likely next words are filtered, and the probability mass is redistributed among only those K next words
Top-p sampling	Exclude low probability tokens - include the most probable tokens, such that the sum of the most probable tokens just reaches p. In other words, based on a parameter $0 <= p <= 1$, Top-P Sampling aggregates the smallest set of words that have summed probability mass p
Repetition_penalty	The more a token is used the more it is penalized to not be picked in successive generation passes



DIY - HuggingFace “GitHub of Machine Learning”

Develop open-source tools for transfer learning in NLP

- Download model
 - <https://huggingface.co/EleutherAI>
 - <https://huggingface.co/nirajsaran/AdTextGenerator>
- Fine tune
 - Python wrapper aitextgen for fine-tuning using your task specific dataset
- Upload
 - huggingface.co/new
- Customize
 - model params: temperature, Top K, num return sequences
 - model card: title text, examples input prompt
- Let others use or fine-tune it for downstream apps!

Super easy to use industry leading models

```
1. from transformers import pipeline  
2. gpt_neo_generator = pipeline('text-generation', model='EleutherAI/gpt-neo-1.3B')  
3. gpt_neo_generator("Best deals on Samsung")
```



Let's see it in action - demos and examples

- GPT-3: <https://beta.openai.com/playground>
- EleutherAI: [text generation testing UI](#)
- Startups applying these models: <https://writesonic.com>

Most biased adjectives/adverbs for each gender

Male	Female
general	little
military	married
united	sexual
political	happy
federal	young
great	soft
national	hot
guilty	tiny
criminal	older
former	black
republican	emotional
american	worried
major	nice
such	live
offensive	lesbian

Concerns:
Bias
Abuse

Google PaLM: high-quality explanation for novel jokes

Input: I tried 10,000 random restarts of my neural network, but I was accused of overfitting. I guess no good seed goes unpunished.

Model Output: This joke is a pun. A neural network is a computer program that can learn from data. A "seed" is a number that is used to initialize a random number generator. A "good seed" is a number that produces a good random number generator. The phrase "no good deed goes unpunished" means that if you do something good, you will be punished for it.

Code Generation

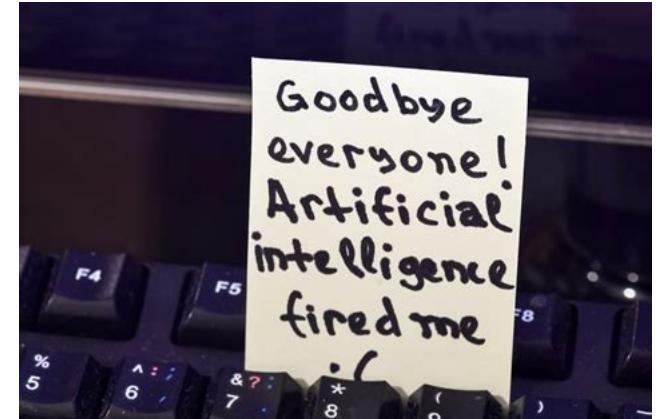
`def factorial(n):`

`def isPalindrome(s):`

`def is_cake(cake):
 """Check whether the cake is true"""`

Summary

- Will AI make coders, journalists, content creators redundant?
- Large Learning Models are becoming bigger, faster and easier to use
- They generate incredible content - no training required
-  is your friend
- Not all the providers provide access to the latest models
- Many startups are making good money with creative applications - Blog writing, Google/Facebook Ads generation, Quora Answers, Landing Pages, LinkedIn posts, YouTube titles, Press Releases for content creators/freelancers



References

- [GPT-3 - Wikipedia](#)
- [Transformer \(machine learning model\) - Wikipedia](#)
- <https://pile.eleuther.ai/>
- <https://arxiv.org/pdf/2101.00027v1.pdf>
- <https://arxiv.org/pdf/2005.14165.pdf>
- [Google AI Blog: Pathways Language Model \(PaLM\): Scaling to 540 Billion Parameters for Breakthrough Performance \(googleblog.com\)](#)