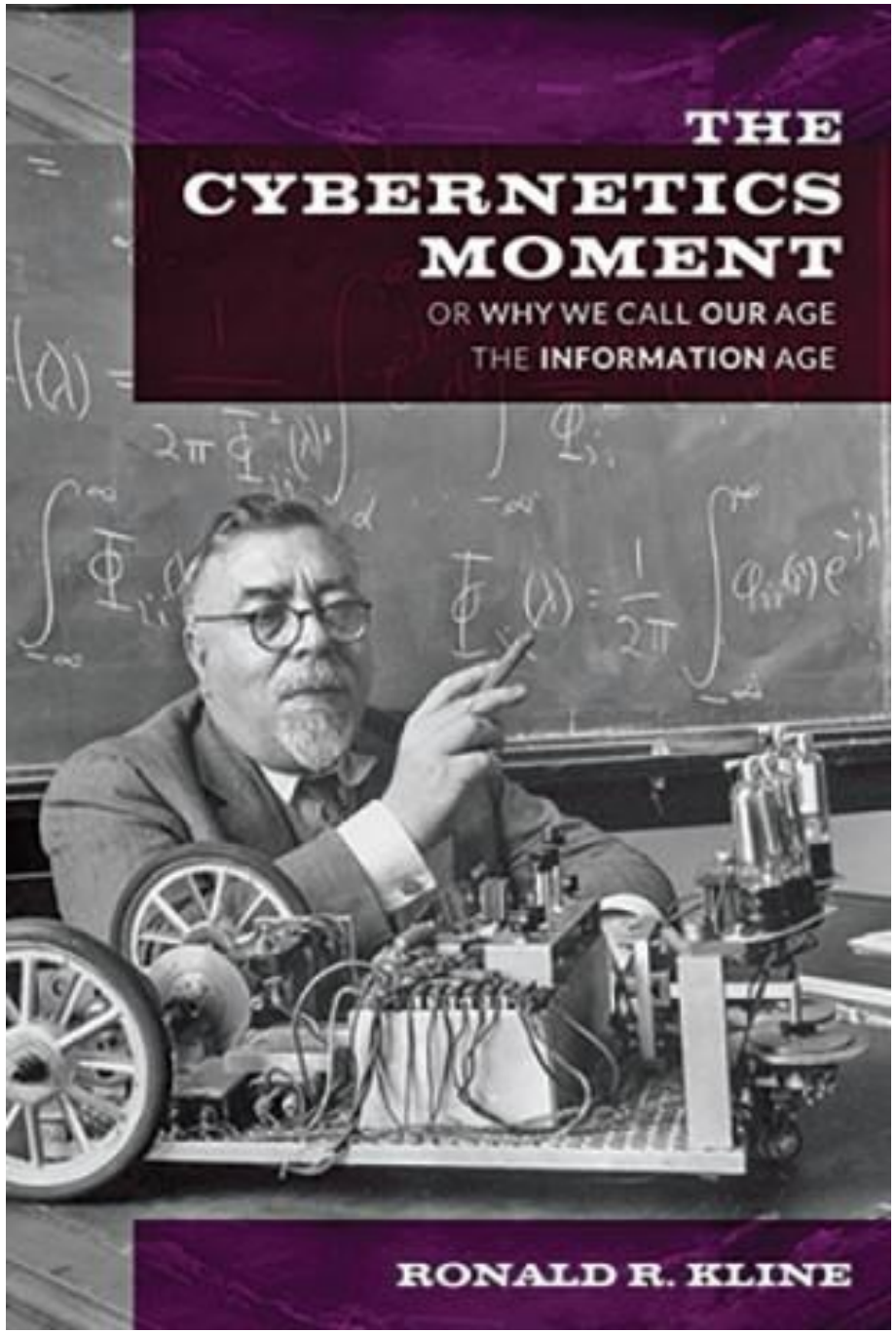
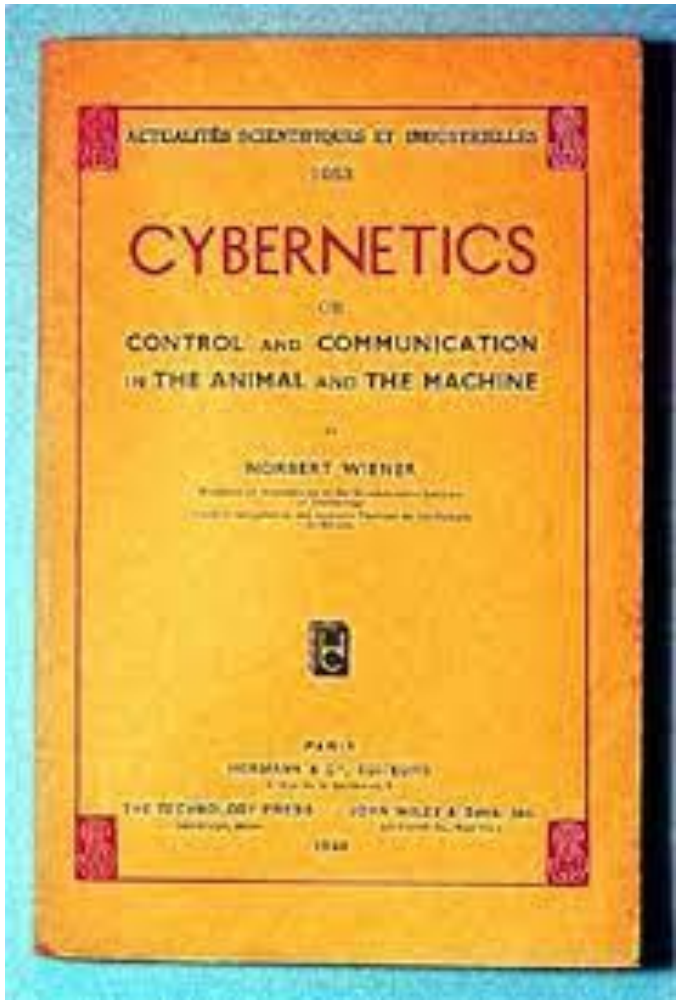
Generative AI Models
ECE 598 LV – Lecture 12

Lav R. Varshney

1 March 2022


THE
CYBERNETICS
MOMENT
OR WHY WE CALL OUR AGE
THE INFORMATION AGE
RONALD R. KLINE


ACTUALITÉS SCIENTIFIQUES ET INDUSTRIELLES
CYBERNETICS
OR
CONTROL AND COMMUNICATION
IN THE ANIMAL AND THE MACHINE
NORBERT WIENER
PARIS
HERMANN & Cie, Editeurs
THE TECHNOLOGY PRESS
JOHN WILEY & SONS, Inc.
1948

5. First-order word approximation. Rather than continue with tetragram, . . . , n-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation. The word transition probabilities are correct but no further structure is included.

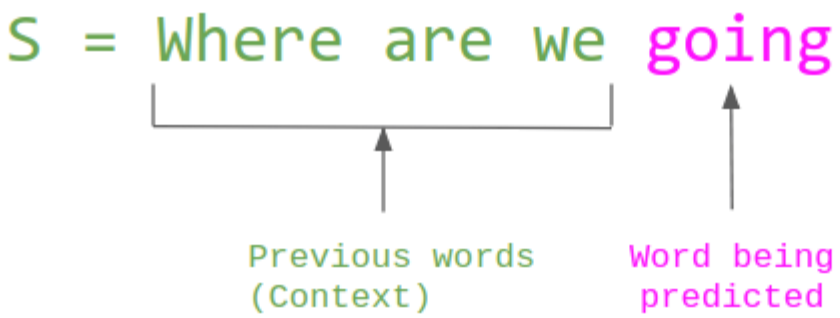
   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

The resemblance to ordinary English text increases quite noticeably at each of the above steps. Note that these samples have reasonably good structure out to about twice the range that is taken into account in their construction. Thus in (3) the statistical process insures reasonable text for two-letter sequences, but four-letter sequences from the sample can usually be fitted into good sentences. In (6) sequences of four or more words can easily be placed in sentences without unusual or strained constructions. The particular sequence of ten words "attack on an English writer that the character of this" is not at all unreasonable. It appears then that a sufficiently complex stochastic process will give a satisfactory representation of a discrete source.

(Shannon, 1948)

Language Model

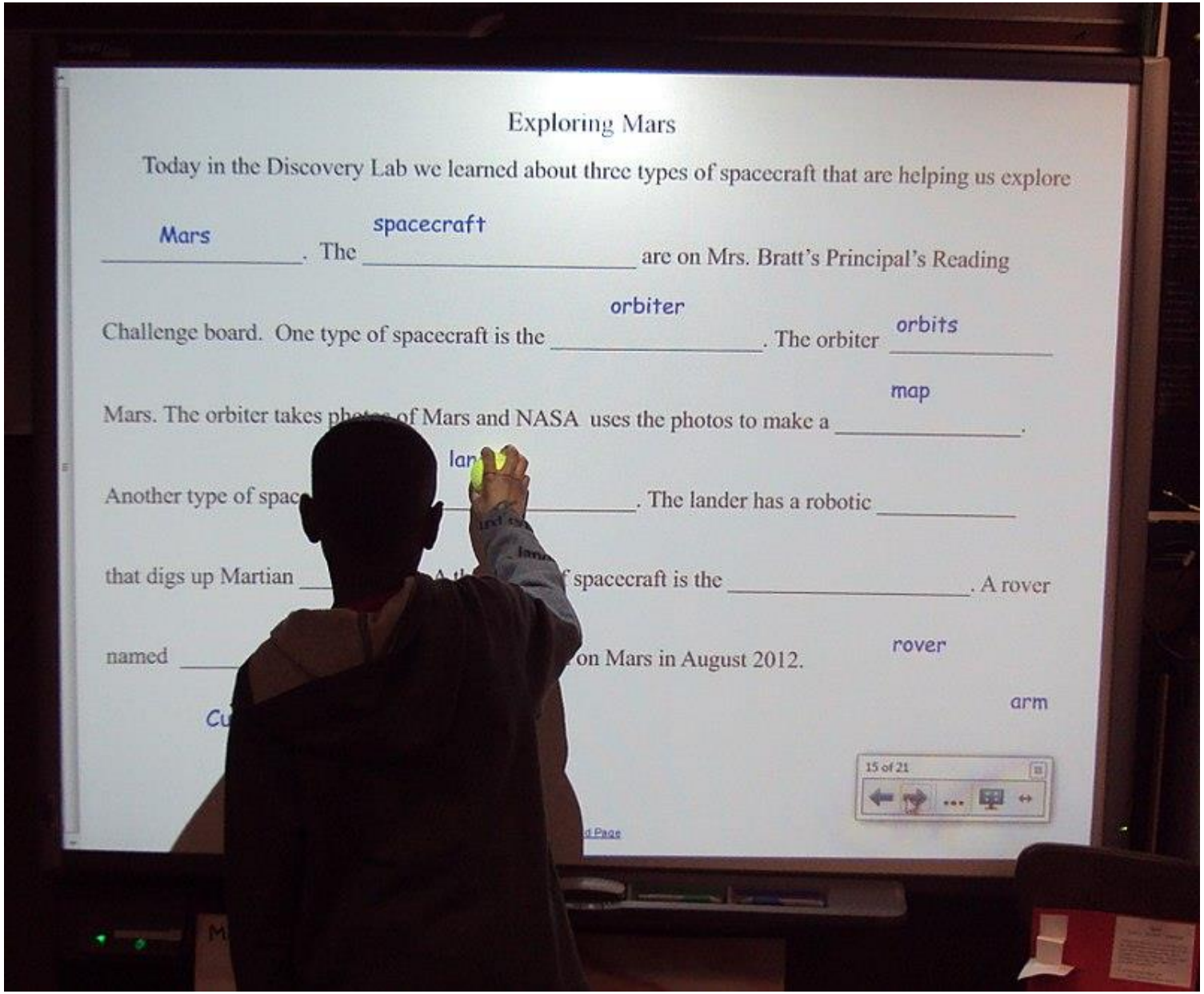
$$P_{(w_1,w_2,\ldots,w_n)} = p(w_1)p(w_2|w_1)p(w_3|w_1,w_2)\ldots p(w_n|w_1,w_2,..,w_{n-1})$$
$$= \prod_{i=1}^{n} p(w_i|w_1,\ldots,w_{i-1}) \quad (1)$$



P(S) = P(Where) x P(are | Where) x P(we | Where are) x P(going | Where are we)

https://thegradient.pub/understanding-evaluation-metrics-for-language-models/

Self-Supervision and Cloze Task



https://transformer.huggingface.co/

CTRL: A Conditional Transformer Language Model for Controllable Generation

Nitish Shirish Keskar[*], **Bryan McCann**[*], **Lav R. Varshney**, **Caiming Xiong**, **Richard Socher**
Salesforce Research[†]

Abstract

Large-scale language models show promising text generation capabilities, but users cannot easily control particular aspects of the generated text. We release CTRL, a 1.63 billion-parameter conditional transformer language model, trained to condition on control codes that govern style, content, and task-specific behavior. Control codes were derived from structure that naturally co-occurs with raw text, preserving the advantages of unsupervised learning while providing more explicit control over text generation. These codes also allow CTRL to predict which parts of the training data are most likely given a sequence. This provides a potential method for analyzing large amounts of data via model-based source attribution. We have released multiple full-sized, pretrained versions of CTRL at `https://github.com/salesforce/ctrl`.

Conditional Generation (CTRL Codes)

Links https://www.cnn.com/2007/09/20/us-president-meets-british-pm
LONDON, England (CNN) – U.S. President George W. Bush met with British Prime Minister Tony Blair on Monday to discuss the war in Iraq, according to a statement from Blair's office.\n\nThe meeting was held at 10 Downing Street and lasted about an hour.\n\n"The two leaders discussed Iraq and other international issues of mutual concern," said Blair.\n\nBush, who is scheduled to meet Wednesday with Russian President Vladimir Putin, will also visit Germany for talks later this week.\n\nIn his statement, Blair said, "We agreed that we should continue our efforts together to bring peace and stability to Iraq. We both reaffirmed our commitment to working closely together, as well as to continuing to work constructively toward achieving lasting security and prosperity throughout the Middle East region."\n\nBush's trip comes after he visited Britain last week where he spoke out against terrorism while visiting Buckingham Palace.\n\nHe has been criticized by some lawmakers over what they say are insufficient military resources being devoted to fighting terrorism.

Data

We train on 140 GB of text drawing from a wide variety of domains: Wikipedia (En, De, Es, Fr), Project Gutenberg[1], submissions from 45 subreddits, OpenWebText[2], a large collection of news data (Hermann et al., 2015; Barrault et al., 2019; Sandhaus, 2008; Grusky et al., 2018), Amazon Reviews (McAuley et al., 2015), Europarl and UN data from WMT (En-De, En-Es, En-Fr) (Barrault et al., 2019), question-answer pairs (no context documents) from ELI5 (Fan et al., 2019) and the MRQA shared task[3], which includes the Stanford Question Answering Dataset (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2016), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), and Natural Questions (Kwiatkowski et al., 2019). A full account of training data and associated control codes can be found in Table 7 in the Appendix.

[1]We use a modified version of `https://github.com/chiphuyen/lazynlp`
[2]We use a modified version of `https://github.com/jcpeterson/openwebtext.git`
[3]`https://github.com/mrqa/MRQA-Shared-Task-2019`

Tokenization

We learn BPE (Sennrich et al., 2015) codes and tokenize the data using fastBPE[4], but we use a large vocabulary of roughly 250K tokens. This includes the sub-word tokens necessary to mitigate problems with rare words, but it also reduces the average number of tokens required to generate long text by including most common words. We use English Wikipedia and a 5% split of our collected OpenWebText data for learning BPE codes. We also introduce an `unknown` token so that during preprocessing we can filter out sequences that contain more than 2 unknown tokens. This, along with the compressed storage for efficient training (TFRecords) (Abadi et al., 2016), reduces our training data to 140 GB from the total 180 GB collected. Data was treated as a single stream of tokens with non-domain control codes inserted where appropriate (often at document boundaries).

Tokenization for Controllable Generation

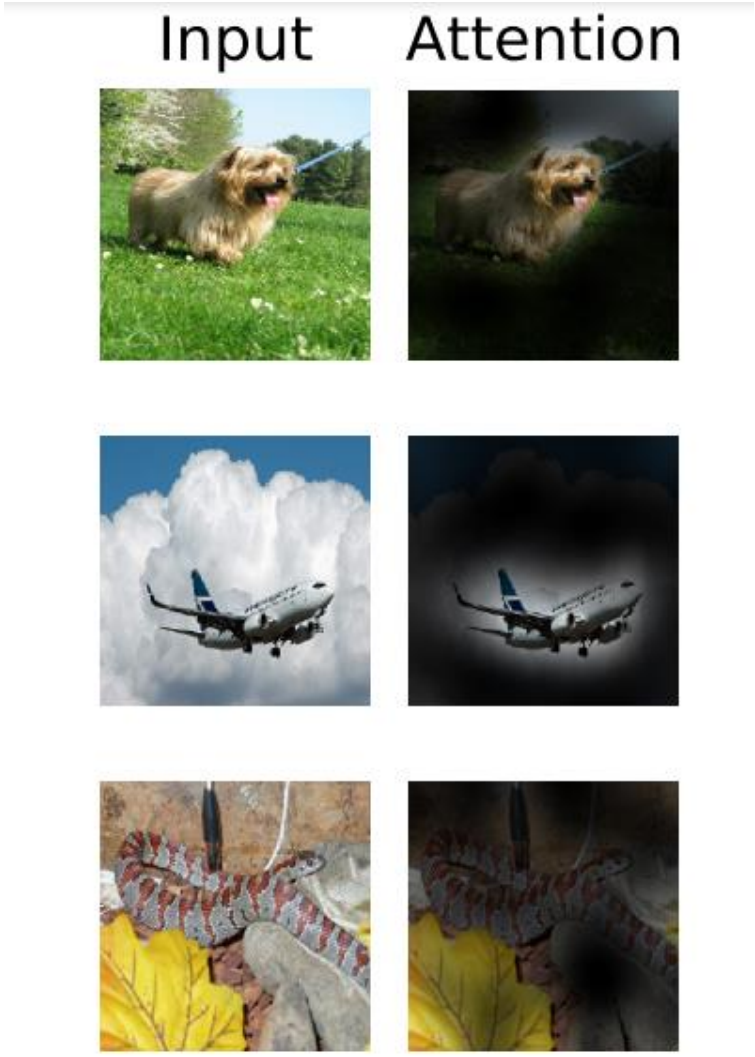
The stream was chunked into contiguous sequences of tokens. Each sequence originated from a domain, and it has the corresponding domain control code prepended as the first token in the sequence. In this way, domain control codes receive special treatment (Kobus et al., 2016). They are propagated to all text in the domain as the first token. This is similar to how codes and natural language sequences have been used in multi-task settings (Wu et al., 2016; Johnson et al., 2017; McCann et al., 2018) to control conditional language models. All other control codes are injected into the data without such special treatment (Moryossef et al., 2019; Caswell et al., 2019). We experimented with sequence lengths of 256 and 512 due to memory and optimization constraints. Despite training on relatively short sequences compared to other approaches, we found that a sliding-window approach allows for generation beyond these windows, and we also found little difference in quality between the two models within the first 256 tokens. Further, we note that our vocabulary is approximately 4 times larger than similar approaches, hence the effective sequence length in characters is comparable.

Architecture and Training Algorithms/Infrastructure

CTRL has model dimension $d = 1280$, inner dimension $f = 8192$, 48 layers, and 16 heads per layer. Dropout with probability 0.1 follows the residual connections in each layer. Token embeddings were tied with the final output embedding layer (Inan et al., 2016; Press & Wolf, 2016).

CTRL was implemented in TensorFlow (Abadi et al., 2016) and trained with a global batch size of 1024 distributed across 256 cores of a Cloud TPU v3 Pod for 800k iterations. Training took approximately 2 weeks using Adagrad (Duchi et al., 2011) with a linear warmup from 0 to 0.05 over 25k steps. The norm of gradients were clipped to 0.25 as in (Merity et al., 2017). Learning rate decay was not necessary due to the monotonic nature of the Adagrad accumulator. We compared to the Adam optimizer (Kingma & Ba, 2014) while training smaller models, but we noticed comparable convergence rates and significant memory savings with Adagrad. We also experimented with explicit memory-saving optimizers including SM3 (Anil et al., 2019), Adafactor (Shazeer & Stern, 2018), and NovoGrad (Ginsburg et al., 2019) with mixed results.

Attention Mechanisms



https://towardsdatascience.com/deconstructing-bert-distilling-6-patterns-from-100-million-parameters-b49113672f77

Attention in Sequence-to-Sequence Models, e.g. for Translation

https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Attention in the Transformer Architecture

https://jalammar.github.io/illustrated-transformer/

Transformers as Universal over Domains

https://www.youtube.com/watch?v=Elxn8rS88bI

Transformers as Universal Predictors?

1258 IEEE TRANSACTIONS ON INFORMATION THEORY VOL. 38, NO. 4, JULY 1992

Universal Prediction of Individual Sequences

Meir Feder, *Member, IEEE,* Neri Merhav, *Member, IEEE,* and
Michael Gutman, *Member, IEEE*

1506 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 50, NO. 7, JULY 2004

Finite-Memory Universal Prediction of Individual Sequences

Eado Meron and Meir Feder, *Fellow, IEEE*