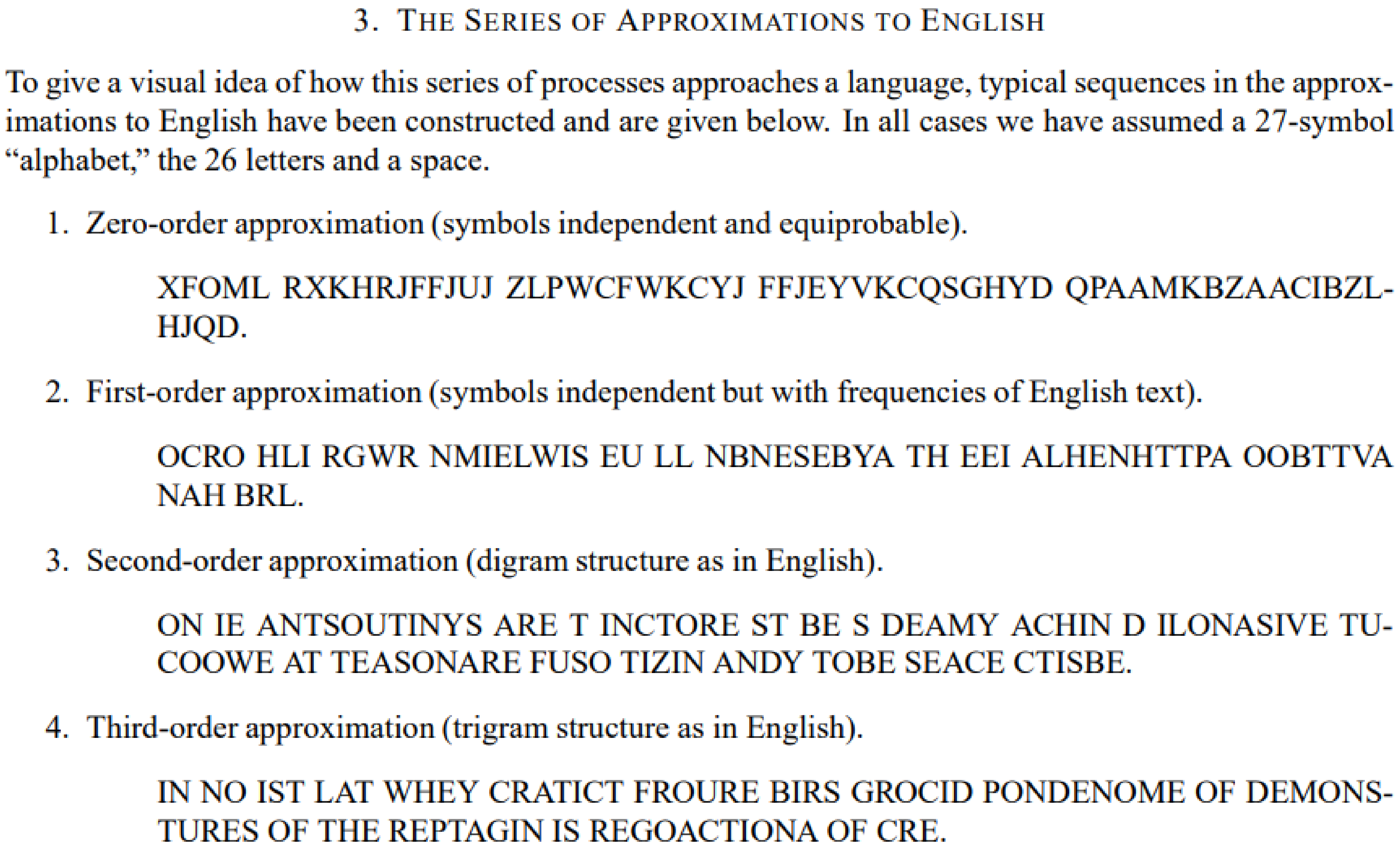
Generative AI Models
ECE 598 LV – Lecture 10

Lav R. Varshney

22 February 2022



3. The Series of Approximations to English

To give a visual idea of how this series of processes approaches a language, typical sequences in the approximations to English have been constructed and are given below. In all cases we have assumed a 27-symbol "alphabet," the 26 letters and a space.

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

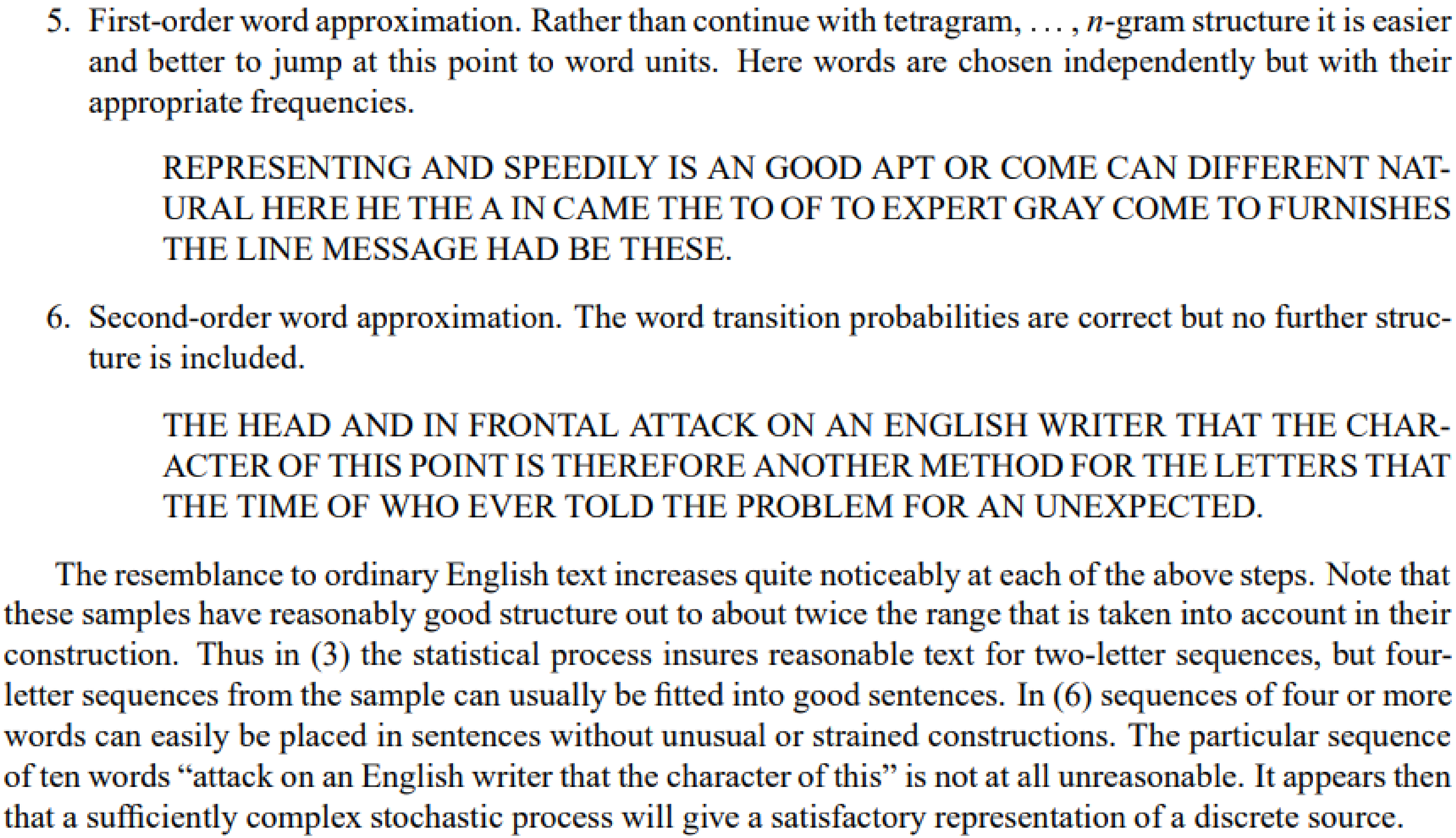
3. Second-order approximation (digram structure as in English).

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

(Shannon, 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.

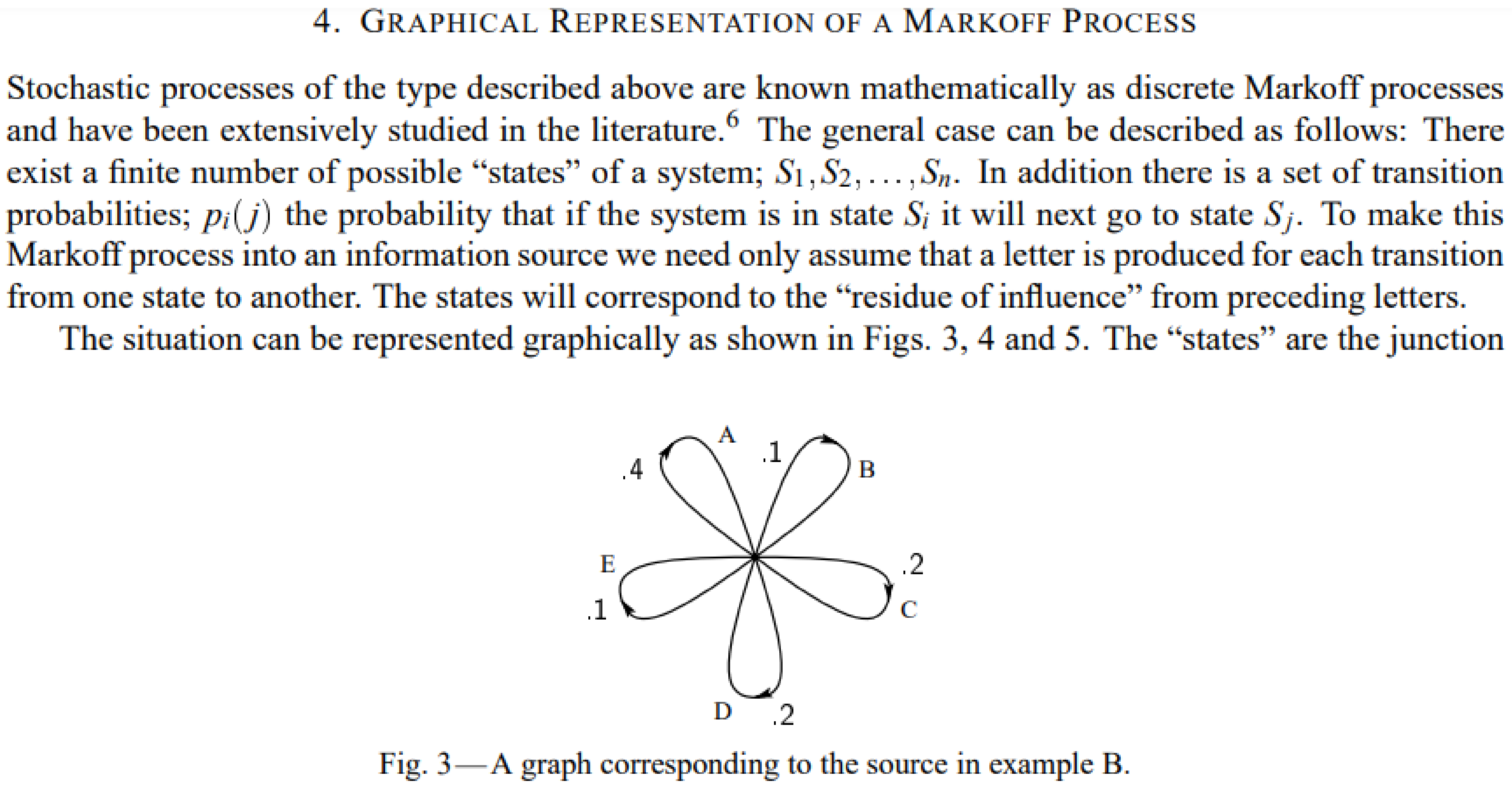
   > REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

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)



4. Graphical Representation of a Markoff Process

Stochastic processes of the type described above are known mathematically as discrete Markoff processes and have been extensively studied in the literature.[6] The general case can be described as follows: There exist a finite number of possible "states" of a system; $S_1, S_2, \ldots, S_n$. In addition there is a set of transition probabilities; $p_i(j)$ the probability that if the system is in state S_i it will next go to state S_j. To make this Markoff process into an information source we need only assume that a letter is produced for each transition from one state to another. The states will correspond to the "residue of influence" from preceding letters.

The situation can be represented graphically as shown in Figs. 3, 4 and 5. The "states" are the junction

Fig. 3—A graph corresponding to the source in example B.

(Shannon, 1948)

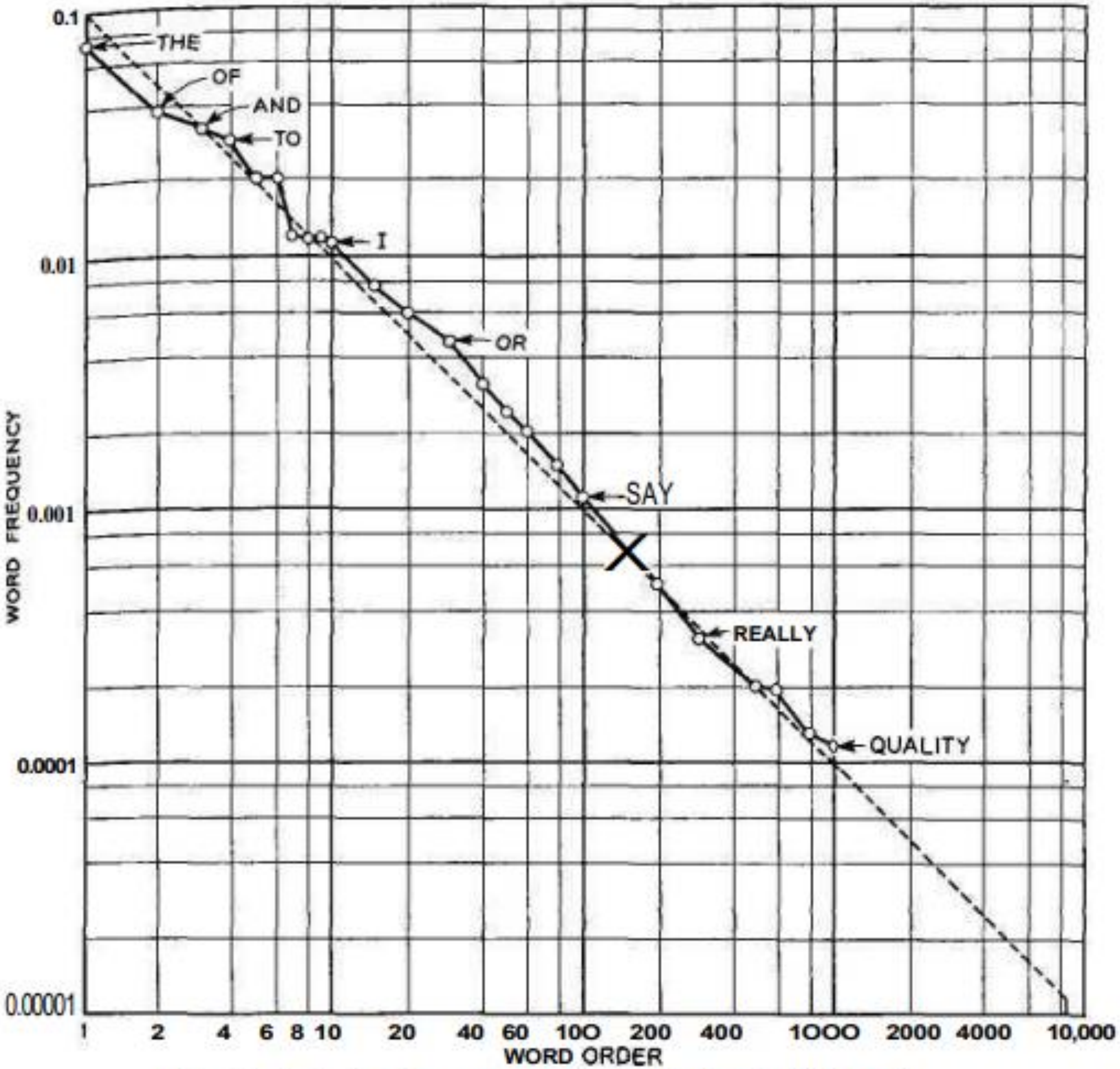
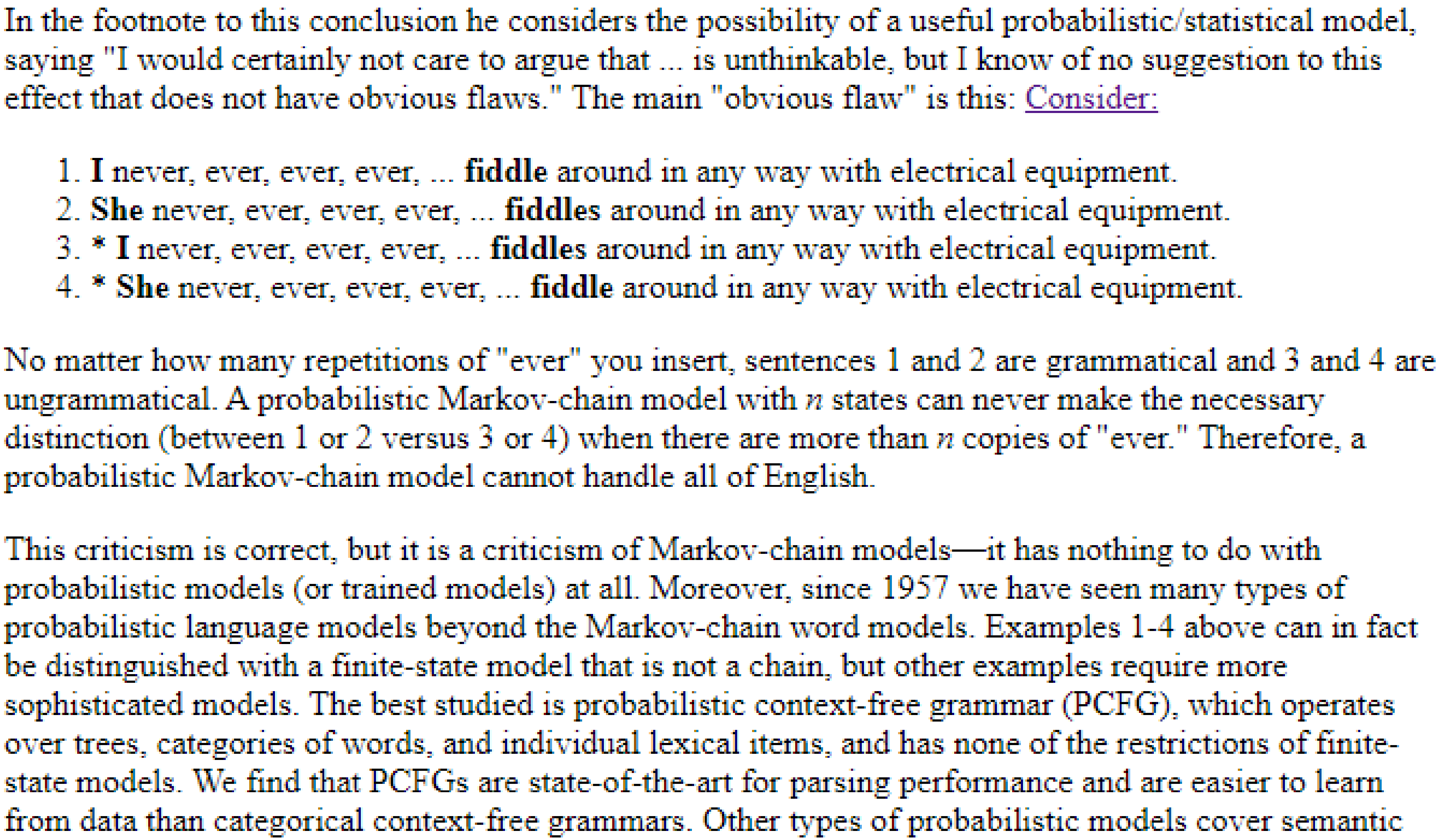


Fig. 1—Relative frequency against rank for English words.

(Shannon, 1951)



In the footnote to this conclusion he considers the possibility of a useful probabilistic/statistical model, saying "I would certainly not care to argue that ... is unthinkable, but I know of no suggestion to this effect that does not have obvious flaws." The main "obvious flaw" is this: Consider:

1. **I** never, ever, ever, ever, ... **fiddle** around in any way with electrical equipment.
2. **She** never, ever, ever, ever, ... **fiddles** around in any way with electrical equipment.
3. * **I** never, ever, ever, ever, ... **fiddles** around in any way with electrical equipment.
4. * **She** never, ever, ever, ever, ... **fiddle** around in any way with electrical equipment.

No matter how many repetitions of "ever" you insert, sentences 1 and 2 are grammatical and 3 and 4 are ungrammatical. A probabilistic Markov-chain model with *n* states can never make the necessary distinction (between 1 or 2 versus 3 or 4) when there are more than *n* copies of "ever." Therefore, a probabilistic Markov-chain model cannot handle all of English.

This criticism is correct, but it is a criticism of Markov-chain models—it has nothing to do with probabilistic models (or trained models) at all. Moreover, since 1957 we have seen many types of probabilistic language models beyond the Markov-chain word models. Examples 1-4 above can in fact be distinguished with a finite-state model that is not a chain, but other examples require more sophisticated models. The best studied is probabilistic context-free grammar (PCFG), which operates over trees, categories of words, and individual lexical items, and has none of the restrictions of finite-state models. We find that PCFGs are state-of-the-art for parsing performance and are easier to learn from data than categorical context-free grammars. Other types of probabilistic models cover semantic

https://norvig.com/chomsky.html

Generative Grammar

येनाक्षरसमाम्नायमधिगम्य महेश्वरात् । कृत्स्नं व्याकरणं प्रोक्तं तस्मै पाणिनये नमः ॥
येन धौता गिरः पुंसां विमलैः शब्दवारिभिः । तमश्चाज्ञानजं भिन्नं तस्मै पाणिनये नमः ॥

Context-free grammar

From Wikipedia, the free encyclopedia

In formal language theory, a **context-free grammar** (**CFG**) is a formal grammar whose production rules are of the form

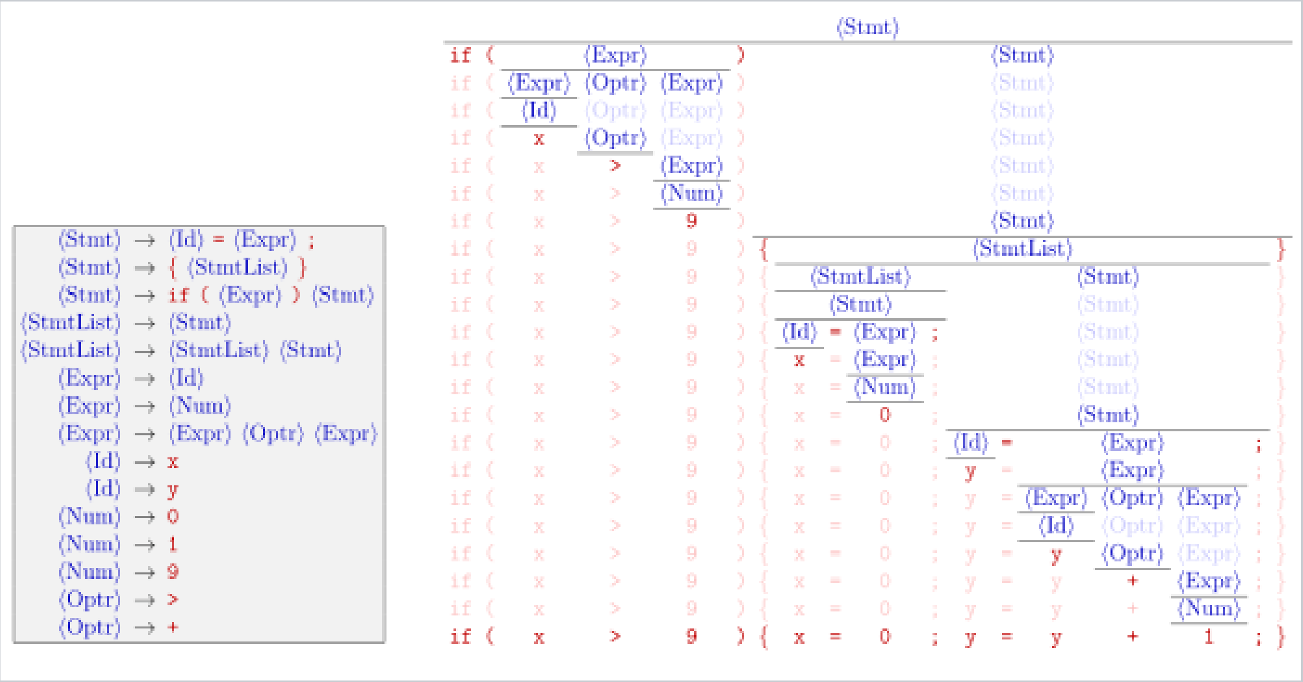
$$A \ \to \ \alpha$$

with A a *single* nonterminal symbol, and α a string of terminals and/or nonterminals (α can be empty). A formal grammar is "context free" if its production rules can be applied regardless of the context of a nonterminal. No matter which symbols surround it, the single nonterminal on the left hand side can always be replaced by the right hand side. This is what distinguishes it from a context-sensitive grammar.

A formal grammar is essentially a set of production rules that describe all possible strings in a given formal language. Production rules are simple replacements. For example, the first rule in the picture,

$$\langle \text{Stmt} \rangle \to \langle \text{Id} \rangle = \langle \text{Expr} \rangle;$$

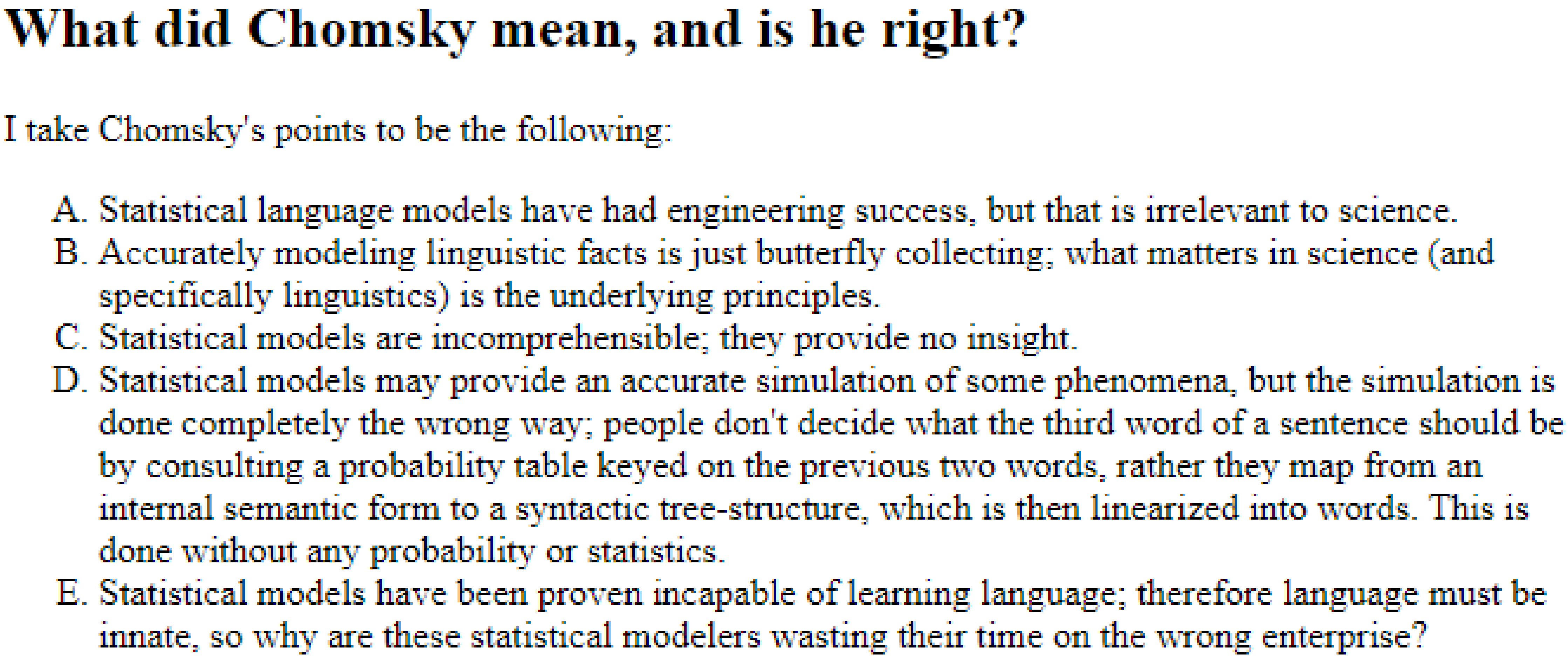
replaces $\langle \text{Stmt} \rangle$ with $\langle \text{Id} \rangle = \langle \text{Expr} \rangle;$. There can be multiple replacement rules for a given nonterminal symbol. The language generated by a grammar is the set of all strings of terminal symbols that can be derived, by repeated rule applications, from some particular nonterminal symbol ("start symbol"). Nonterminal symbols are used during the derivation process, but do not appear in its final result string.



```
    ⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
    ⟨Stmt⟩ → { ⟨StmtList⟩ }
    ⟨Stmt⟩ → if ( ⟨Expr⟩ ) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
    ⟨Expr⟩ → ⟨Id⟩
    ⟨Expr⟩ → ⟨Num⟩
    ⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
      ⟨Id⟩ → x
      ⟨Id⟩ → y
     ⟨Num⟩ → 0
     ⟨Num⟩ → 1
     ⟨Num⟩ → 9
    ⟨Optr⟩ → >
    ⟨Optr⟩ → +
```

```
⟨Stmt⟩
if ( ⟨Expr⟩ ) ⟨Stmt⟩
if ( ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩ ) ⟨Stmt⟩
if ( ⟨Id⟩ ⟨Optr⟩ ⟨Expr⟩ ) ⟨Stmt⟩
if ( x ⟨Optr⟩ ⟨Expr⟩ ) ⟨Stmt⟩
if ( x > ⟨Expr⟩ ) ⟨Stmt⟩
if ( x > ⟨Num⟩ ) ⟨Stmt⟩
if ( x > 9 ) ⟨Stmt⟩
if ( x > 9 ) { ⟨StmtList⟩ }
if ( x > 9 ) { ⟨StmtList⟩ ⟨Stmt⟩ }
if ( x > 9 ) { ⟨Stmt⟩ ⟨Stmt⟩ }
if ( x > 9 ) { ⟨Id⟩ = ⟨Expr⟩ ; ⟨Stmt⟩ }
if ( x > 9 ) { x = ⟨Expr⟩ ; ⟨Stmt⟩ }
if ( x > 9 ) { x = ⟨Num⟩ ; ⟨Stmt⟩ }
if ( x > 9 ) { x = 0 ; ⟨Stmt⟩ }
if ( x > 9 ) { x = 0 ; ⟨Id⟩ = ⟨Expr⟩ ; }
if ( x > 9 ) { x = 0 ; y = ⟨Expr⟩ ; }
if ( x > 9 ) { x = 0 ; y = ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩ ; }
if ( x > 9 ) { x = 0 ; y = ⟨Id⟩ ⟨Optr⟩ ⟨Expr⟩ ; }
if ( x > 9 ) { x = 0 ; y = y ⟨Optr⟩ ⟨Expr⟩ ; }
if ( x > 9 ) { x = 0 ; y = y + ⟨Expr⟩ ; }
if ( x > 9 ) { x = 0 ; y = y + ⟨Num⟩ ; }
if ( x > 9 ) { x = 0 ; y = y + 1 ; }
```

Simplified excerpt of the formal grammar[1] for the C programming language (left), and a derivation of a piece of C code (right) from the nonterminal symbol $\langle \text{Stmt} \rangle$. Nonterminal and terminal symbols are shown in blue and red, respectively.

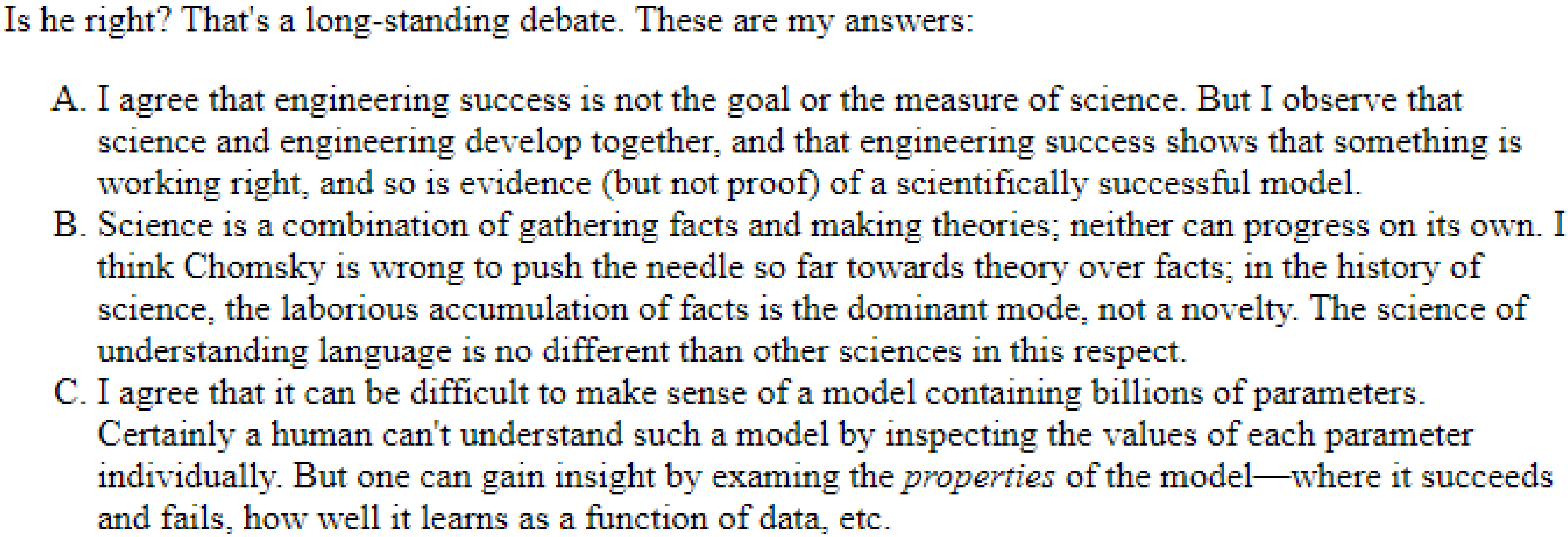


What did Chomsky mean, and is he right?

I take Chomsky's points to be the following:

A. Statistical language models have had engineering success, but that is irrelevant to science.
B. Accurately modeling linguistic facts is just butterfly collecting; what matters in science (and specifically linguistics) is the underlying principles.
C. Statistical models are incomprehensible; they provide no insight.
D. Statistical models may provide an accurate simulation of some phenomena, but the simulation is done completely the wrong way; people don't decide what the third word of a sentence should be by consulting a probability table keyed on the previous two words, rather they map from an internal semantic form to a syntactic tree-structure, which is then linearized into words. This is done without any probability or statistics.
E. Statistical models have been proven incapable of learning language; therefore language must be innate, so why are these statistical modelers wasting their time on the wrong enterprise?

https://norvig.com/chomsky.html



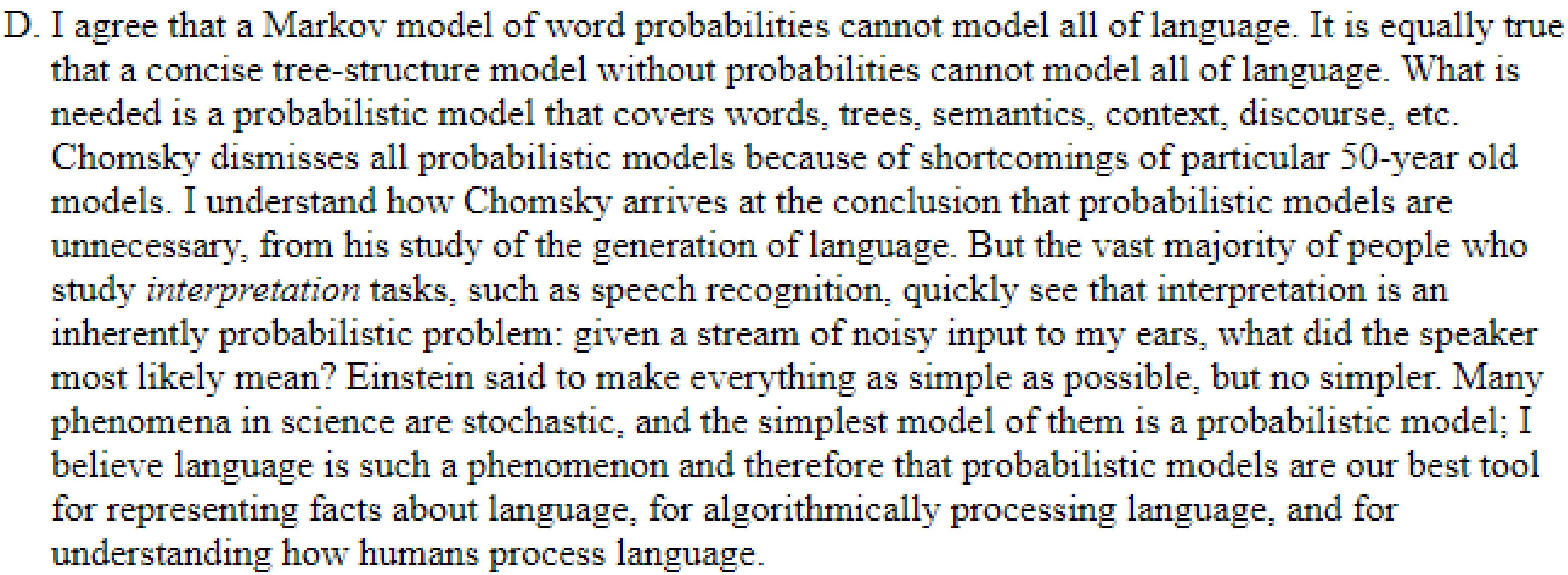
Is he right? That's a long-standing debate. These are my answers:

A. I agree that engineering success is not the goal or the measure of science. But I observe that science and engineering develop together, and that engineering success shows that something is working right, and so is evidence (but not proof) of a scientifically successful model.

B. Science is a combination of gathering facts and making theories; neither can progress on its own. I think Chomsky is wrong to push the needle so far towards theory over facts; in the history of science, the laborious accumulation of facts is the dominant mode, not a novelty. The science of understanding language is no different than other sciences in this respect.

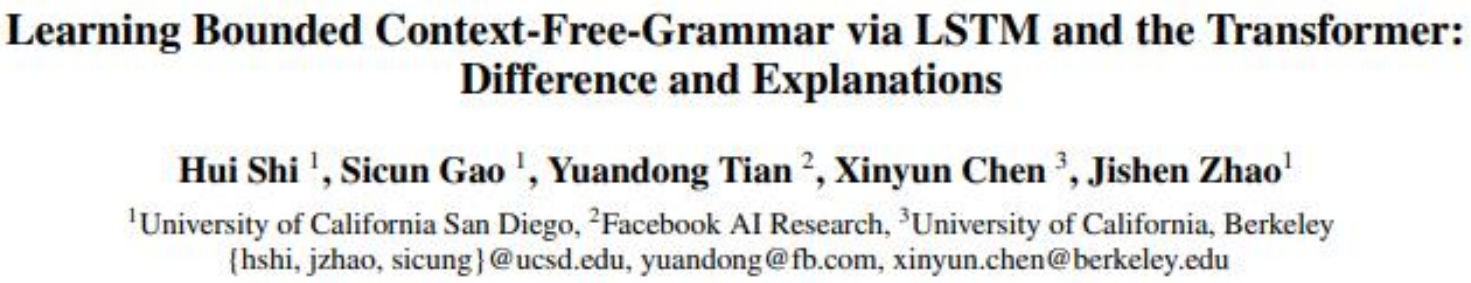
C. I agree that it can be difficult to make sense of a model containing billions of parameters. Certainly a human can't understand such a model by inspecting the values of each parameter individually. But one can gain insight by examing the *properties* of the model—where it succeeds and fails, how well it learns as a function of data, etc.

https://norvig.com/chomsky.html



D. I agree that a Markov model of word probabilities cannot model all of language. It is equally true that a concise tree-structure model without probabilities cannot model all of language. What is needed is a probabilistic model that covers words, trees, semantics, context, discourse, etc. Chomsky dismisses all probabilistic models because of shortcomings of particular 50-year old models. I understand how Chomsky arrives at the conclusion that probabilistic models are unnecessary, from his study of the generation of language. But the vast majority of people who study *interpretation* tasks, such as speech recognition, quickly see that interpretation is an inherently probabilistic problem: given a stream of noisy input to my ears, what did the speaker most likely mean? Einstein said to make everything as simple as possible, but no simpler. Many phenomena in science are stochastic, and the simplest model of them is a probabilistic model; I believe language is such a phenomenon and therefore that probabilistic models are our best tool for representing facts about language, for algorithmically processing language, and for understanding how humans process language.

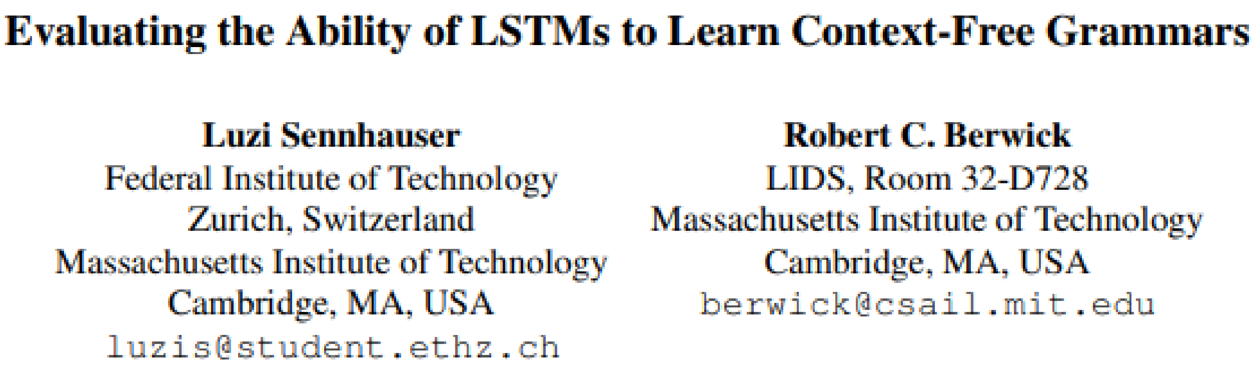
https://norvig.com/chomsky.html



Learning Bounded Context-Free-Grammar via LSTM and the Transformer: Difference and Explanations

Hui Shi [1], Sicun Gao [1], Yuandong Tian [2], Xinyun Chen [3], Jishen Zhao[1]

[1]University of California San Diego, [2]Facebook AI Research, [3]University of California, Berkeley
{hshi, jzhao, sicung}@ucsd.edu, yuandong@fb.com, xinyun.chen@berkeley.edu

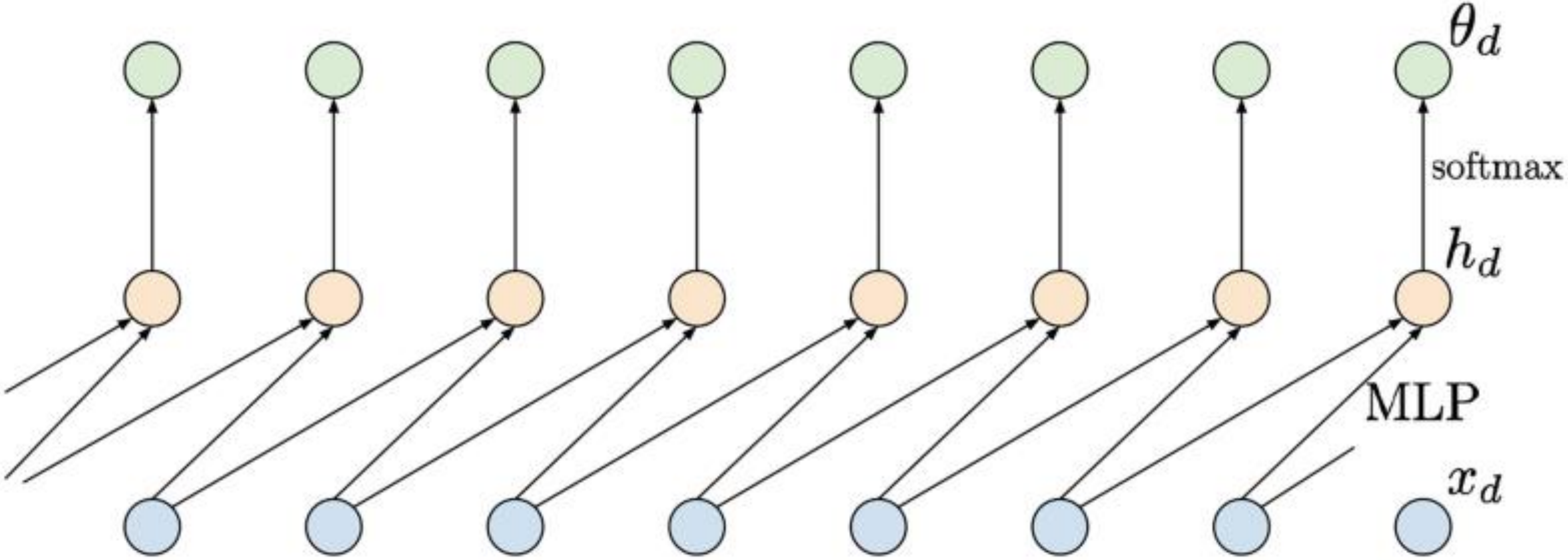


Evaluating the Ability of LSTMs to Learn Context-Free Grammars

Luzi Sennhauser
Federal Institute of Technology
Zurich, Switzerland
Massachusetts Institute of Technology
Cambridge, MA, USA
luzis@student.ethz.ch

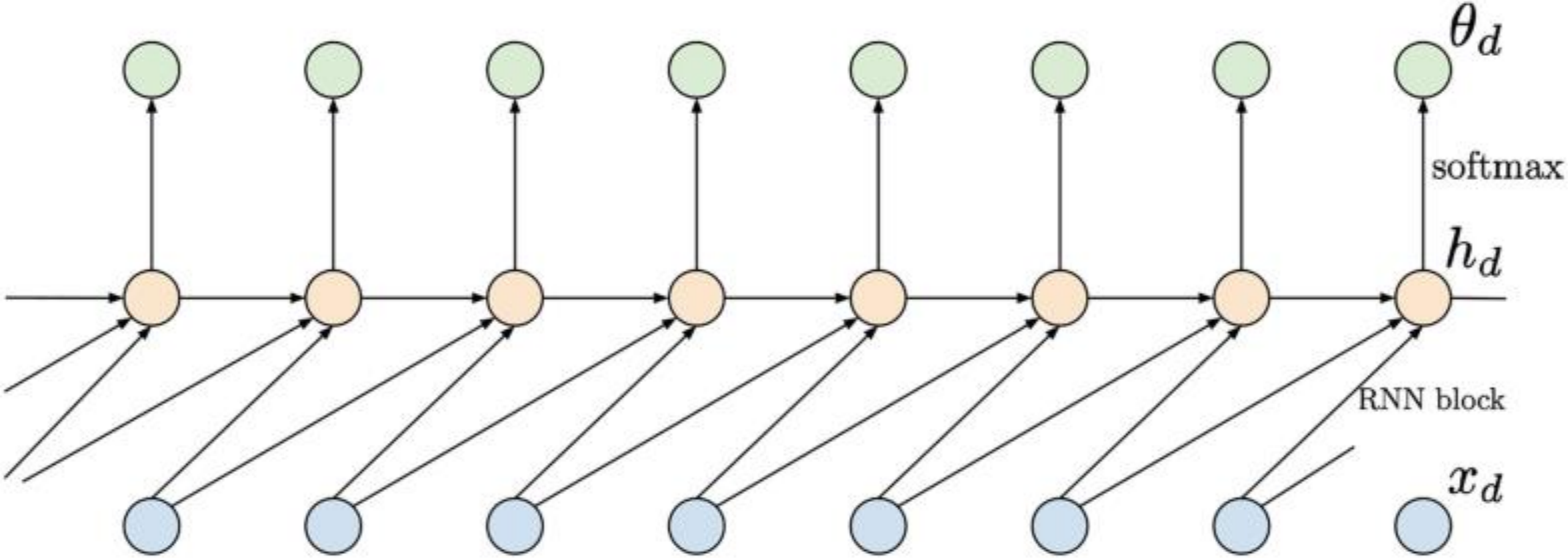
Robert C. Berwick
LIDS, Room 32-D728
Massachusetts Institute of Technology
Cambridge, MA, USA
berwick@csail.mit.edu

Can one learn a language model for a (probabilistic) context-free grammar source and do information-theoretic probing of what rules are learned?



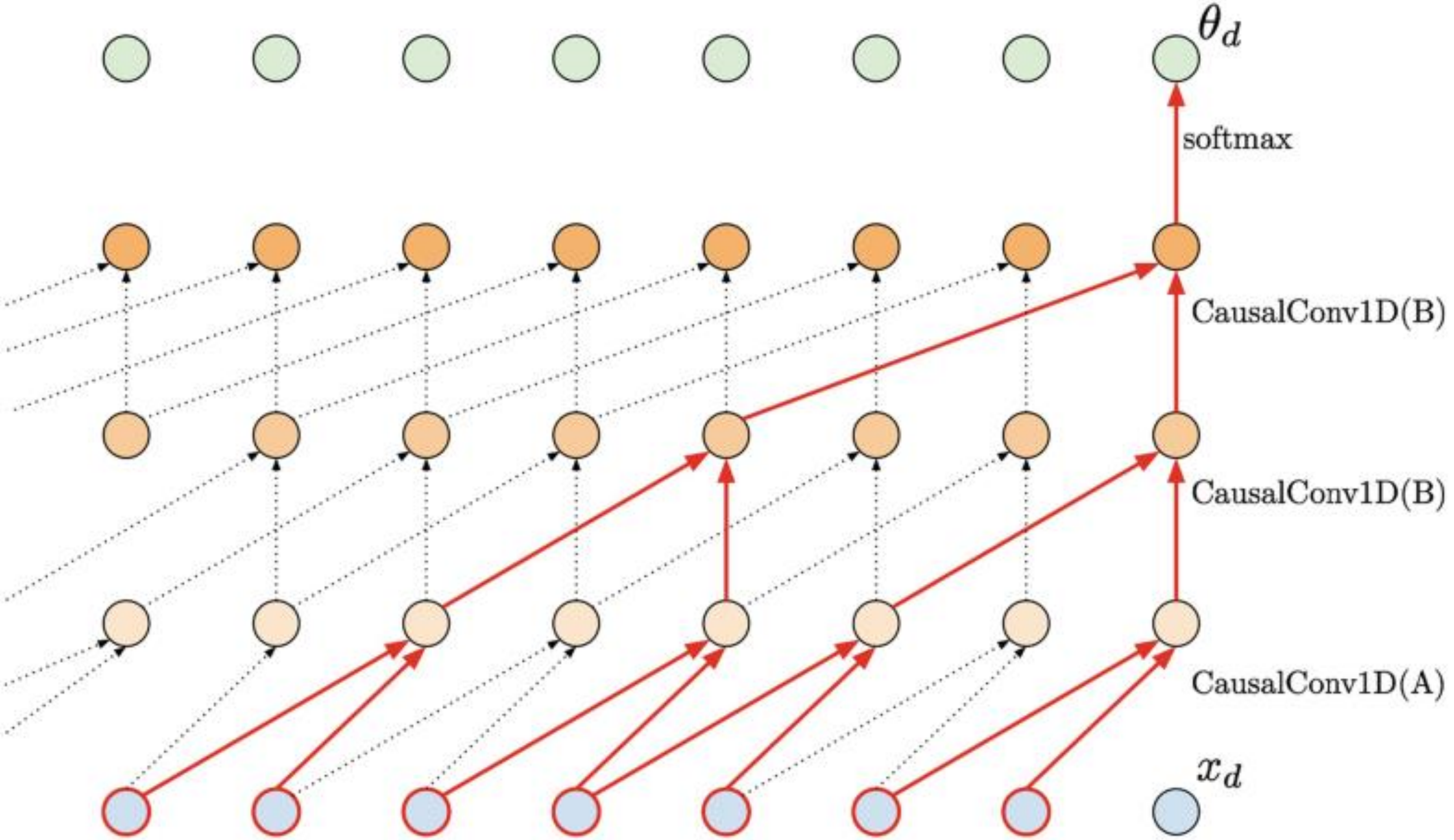
An example of applying a shared MLP depending on two last inputs. Inputs are denoted by blue nodes (bottom), intermediate representations are denoted by orange nodes (middle), and output probabilities are denoted by green nodes (top). Notice that a probability θ_d is not dependent on x_d

[J. M. Tomczak, *Deep Generative Modeling*, Springer, 2022.]



An example of applying an RNN depending on two last inputs. Inputs are denoted by blue nodes (bottom), intermediate representations are denoted by orange nodes (middle), and output probabilities are denoted by green nodes (top). Notice that compared to the approach with a shared MLP, there is an additional dependency between intermediate nodes h_d

[J. M. Tomczak, *Deep Generative Modeling*, Springer, 2022.]



An example of applying causal convolutions. The kernel size is 2, but by applying dilation in higher layers, a much larger input could be processed (red edges), thus, a larger memory is utilized. Notice that the first layers must be option A to ensure proper processing

[J. M. Tomczak, *Deep Generative Modeling*, Springer, 2022.]