A QUANTUM SEARCH DECODER FOR NATURAL LANGUAGE PROCESSING

JOHANNES BAUSCH CAMBRIDGE
SATHYA SUBRAMANIAN CAMBRIDGE
STEPHEN PIDDOCK BRISTOL

QuID
ZÜRICH, 12.9.2019

Quantum Search

Grover, Maximum Finding, OAA
Search with Advice

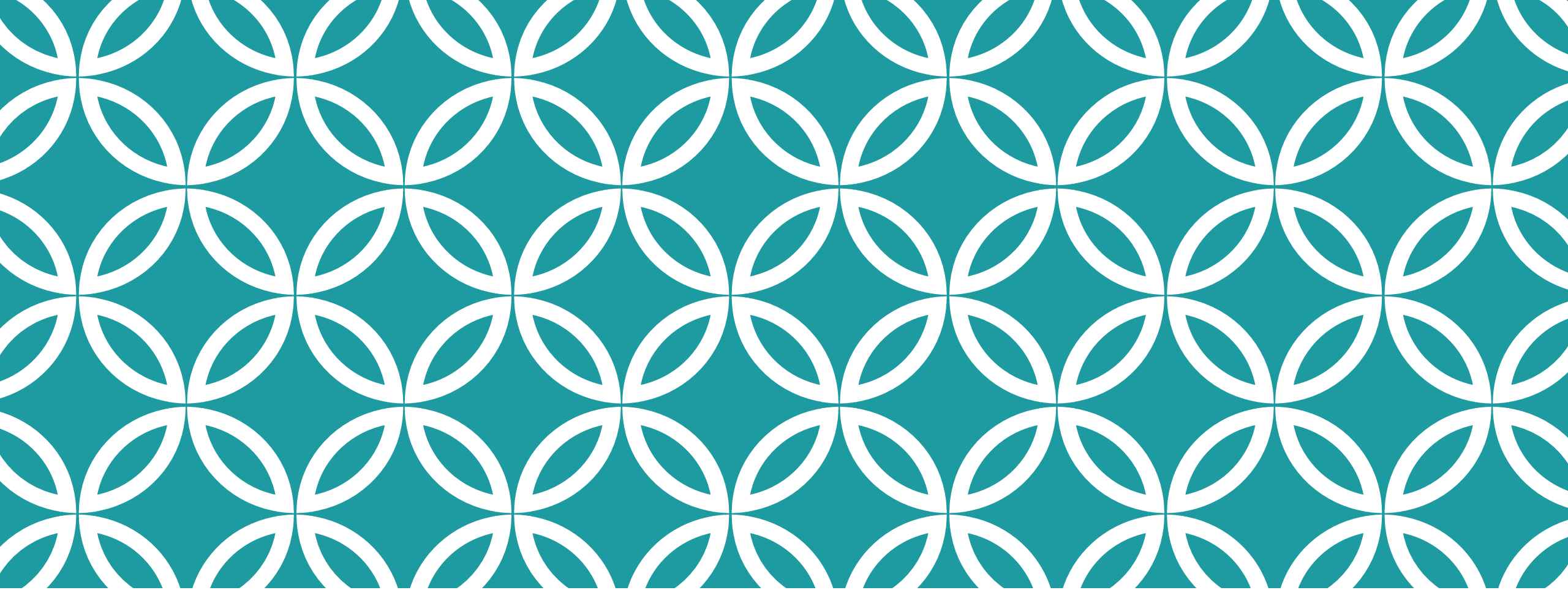
Natural Language Processing

Formal Language Parsing, Generative Models
DeepSpeech: an LSTM for Speech Recognition

Quantum Search Decoder

Beam Search
Advice for Language Decoding
Decoding Algorithm and Runtime Analysis

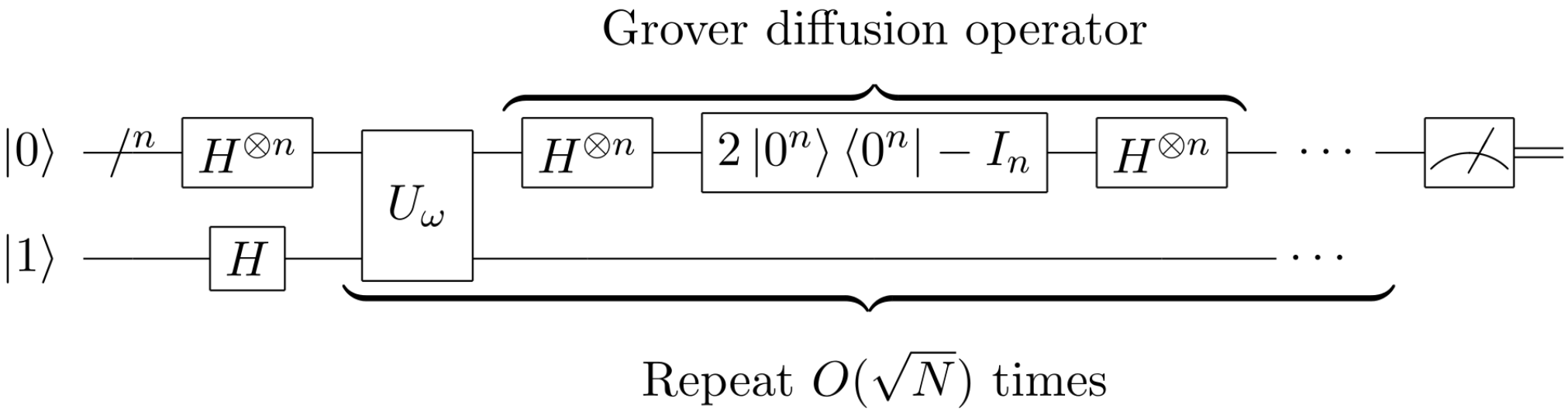
A QUANTUM SEARCH DECODER FOR NATURAL LANGUAGE PROCESSING



QUANTUM SEARCH

Grover
Maximum Finding
Advice Oracles

QUANTUM SEARCH



QUANTUM SEARCH

1. Yields (at least) the well-known quadratic speedup
2. Building block of several quantum algorithms

    1. Repeat-until-success circuits: FIXED POINT OBLIVIOUS AMPLIFICATION
    Guerreschi ’18, Svore et al. ’13

    $$\mathbf{W}|0\rangle|\psi\rangle = \sqrt{p}|0\rangle|\psi'\rangle + \sqrt{1-p}|\perp\rangle$$

    2. Hamiltonian simulation: FUNCTIONS OF UNITARIES, LINEAR COMBINATIONS
    Berry/Childs et al. ’17

    $$\mathbf{W}|\psi\rangle = \sqrt{p}\sum_i \beta_i \mathbf{V}_i|\psi\rangle + \sqrt{1-p}|\perp\rangle$$

    3. Speeding up CSPs: TRAVELLING SALESMAN, GRAPH COLORING
    Moylett/Linden/Montanaro ’18

QUANTUM MAXIMUM FINDING

$$\mathrm{O}\left(\sqrt{\frac{n}{k}}\right)$$

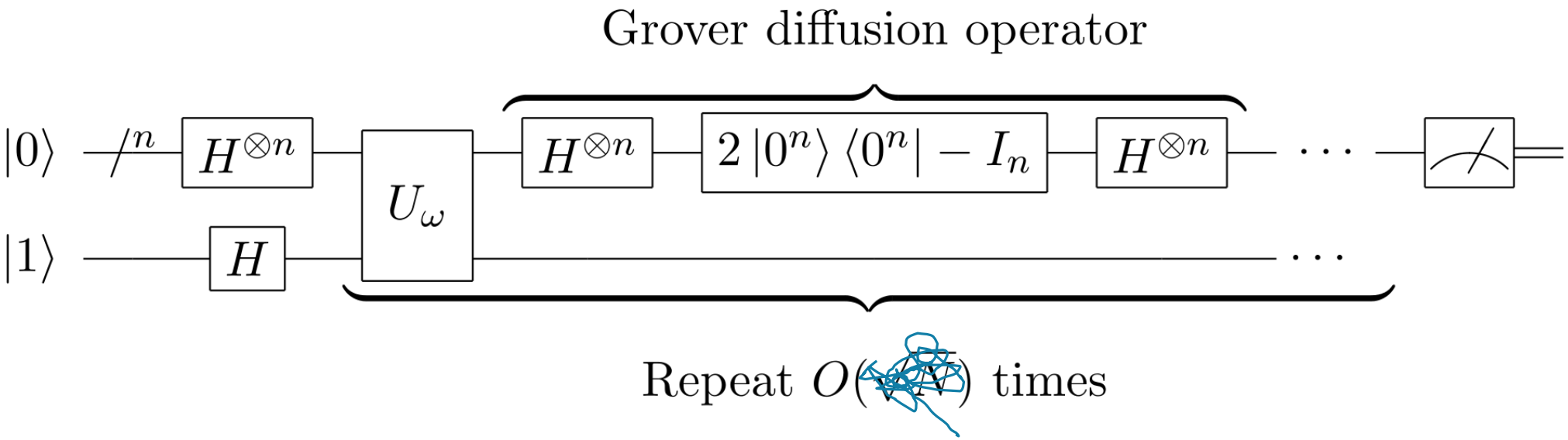
Expected complexity

function FINDMAXIMUM$_m(f, n)$
  $piv \leftarrow -\infty$
  $counter \leftarrow 0$
  repeat
    $cmp \leftarrow (\cdot) \mapsto \cdot > piv$ ▷ comparator against current best score
    $|\psi\rangle \leftarrow$ GROVERSEARCH$(cmp \circ f)$ ▷ amplify elements larger than pivot
    $bestIndex \leftarrow \mathbf{M}_{\text{idx}} |\psi\rangle$ ▷ measure index of larger element
    $piv = \max\{piv, f(bestIndex)\}$
    $counter \leftarrow counter + 1$
  until $counter = m$
end function

Uniform superposition

QUANTUM EXPONENTIAL SEARCH

Number of necessary iterations unknown in case of an unknown number of marked elements!

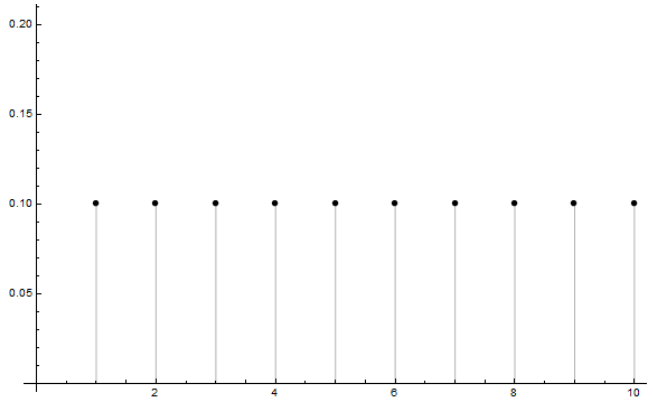


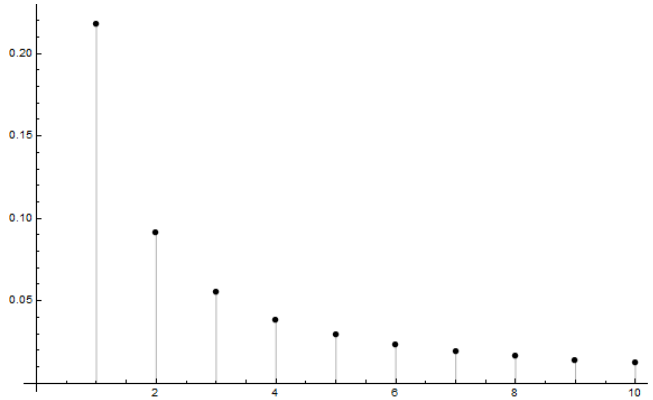
Repeat an exponentially-increasing number of times; overall speedup still quadratic.

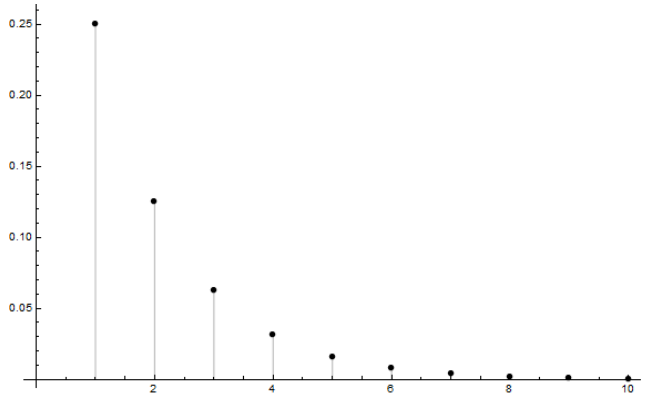
SEARCH WITH ADVICE

Montanaro ’09

Instead of having no prior information about the location of the element to search for, you are given an advice distribution $\mathcal{D}$ with pdf $p(r)$, where r is the rank of the element.







SEARCH WITH ADVICE

Montanaro '09

Instead of having no prior information about the location of the element to search for, you are given an advice distribution μ with pdf $p(r)$, where r is the rank of the element.

Two cases: either μ is *known*, or it is *unknown*.

SEARCH WITH ADVICE: KNOWN DISTRIBUTION

1. The optimal algorithm is clearly to first look at the most likely element, i.e. where p(r) is largest.
2. If r denotes the rank, this is p(0).
3. If not found, keep looking at p(1), p(2), … until the element is found.

Quantumly, you get a quadratic speedup over this classical scheme.

THIS TYPE OF ADVICE IS USEFUL CLASSICALLY.

SEARCH WITH ADVICE: UNKNOWN DISTRIBUTION

1. The only thing we can do is obtain samples.
2. Classically, we have an expected runtime

$$\sum_{r=1}^{n} p(r) \times \frac{1}{p(r)} = n$$

THIS TYPE OF ADVICE IS USELESS CLASSICALLY.

SEARCH WITH ADVICE: UNKNOWN DISTRIBUTION

1. The only thing we can do is obtain samples.
2. On a quantum computer, we can amplify the advice:

$$\sum_{r=1}^{n} p(r) \times \frac{1}{\sqrt{p(r)}} = \sum_{r=1}^{n} \sqrt{p(r)}$$

(almost)

THIS TYPE OF ADVICE IS USELESS CLASSICALLY.
BUT IT CAN BE USEFUL ON A QUANTUM COMPUTER.

SEARCH WITH ADVICE: UNKNOWN DISTRIBUTION

function SEARCHWITHADVICE$_k$($\mathbf{U}_\mu$, f)
  $counter \leftarrow 0$
  repeat
    $guess \leftarrow \mathbf{M}_{\text{idx}}\mathbf{U}_\mu |0\rangle$
    if $guess$ is marked **then**
      return
    end if
    $i \leftarrow \text{Uniform}\{0, \ldots, \lfloor k^j \rfloor - 1\}$
    $|\psi\rangle \leftarrow \text{AMPLITUDEAMPLIFY}(f, i)$ ▷ amplify elements marked by f for i rounds
    $guess \leftarrow \mathbf{M}_{\text{idx}} |\psi\rangle$
    if $guess$ is marked **then**
      return
    end if
    $counter \leftarrow counter + 1$
  until $counter = \lfloor \log_k \sqrt{n} \rfloor$
  return GROVERSEARCH(()f) ▷ fallback: do normal Grover search
end function

$$\mathbf{U}_\mu |0\rangle = \sum_{i=1}^{n} \sqrt{p(i)} |i\rangle$$

▷ early exit for extremely biased μ

SEARCH WITH ADVICE: UNKNOWN DISTRIBUTION

In expectation, need

$$\mathrm{O}\left(\min\left\{\frac{1}{\sqrt{p(x)}}, \sqrt{n}\right\}\right)$$

queries to f and the advice oracle.

$$\text{For a powerlaw, i.e. when } p(r) \propto r^{-k}, \text{ runtime} = \begin{cases} \mathrm{O}(\sqrt{n}) & k \in [-1, 0] \\ \mathrm{O}(n^{-(1/2+1/k)} & k \in (-2, -1) \\ \mathrm{O}(\log n) & k = -2 \\ \mathrm{O}(1) & \text{otherwise.} \end{cases}$$

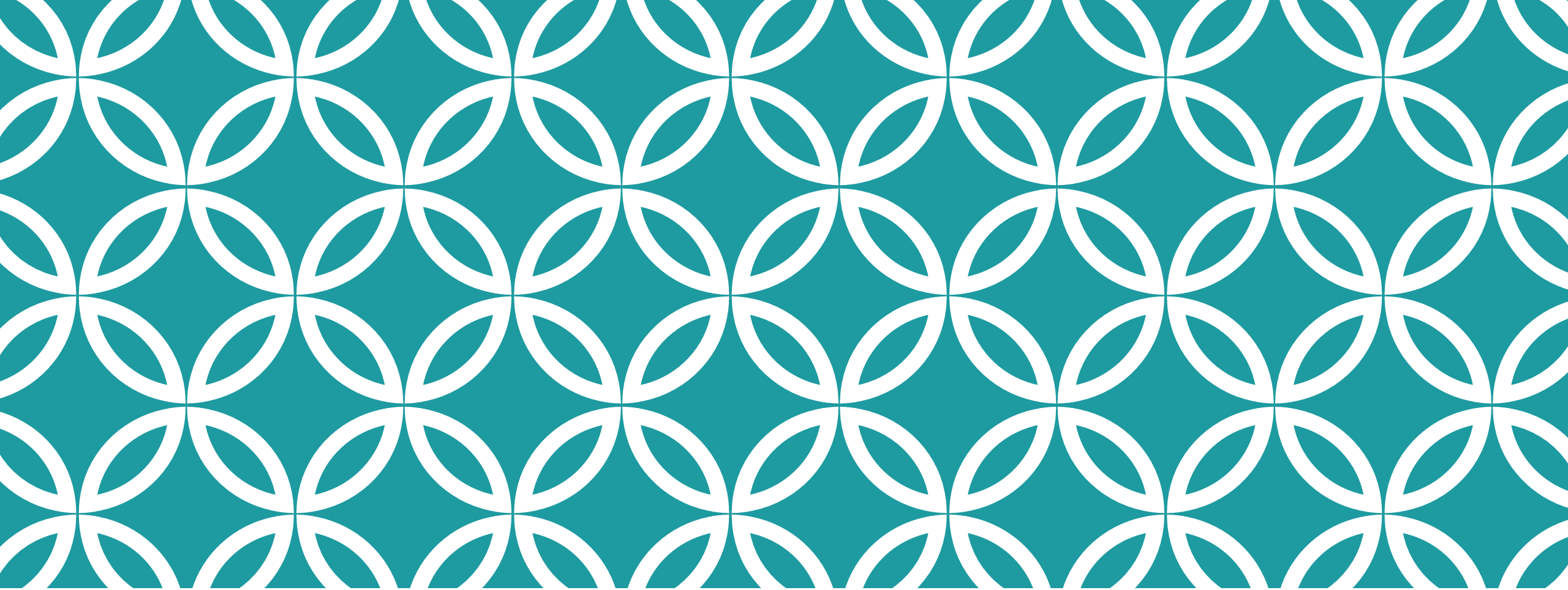
QUANTUM MAXIMUM FINDING WITH ADVICE

Based on a generalized maximum finding routine
VANAPPELDOORN ET AL., '18

and search with advice
MONTANARO '09

Theorem 1. *Given an advice oracle* $\mathbf{U}_\mu$ *with a power law distribution* $\mathrm{pdf}_\mu(r) \propto r^{-k}$, *where* μ *advices on the maximum element, we can find the maximum element in expected time*

$$\begin{cases} \mathrm{O}(\sqrt{n}) & k \in [-1, 0] \\ \mathrm{O}(n^{-(1/2+1/k)} & k \in (-2, -1) \\ \mathrm{O}(\log n) & k = -2 \\ \mathrm{O}(1) & \textit{otherwise.} \end{cases}$$



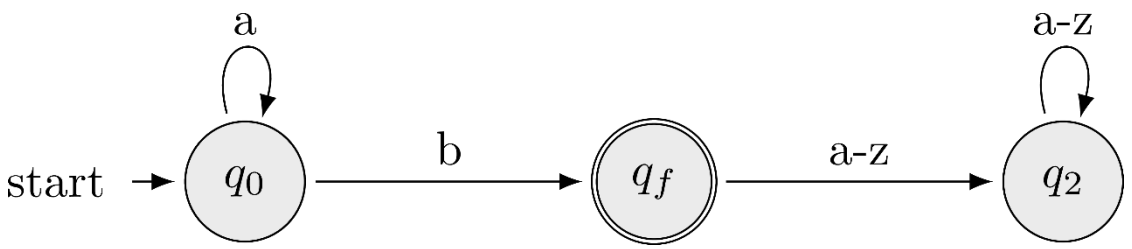
(NATURAL) LANGUAGE PROCESSING

Formal Languages
Parsing
Generative Models
DeepSpeech

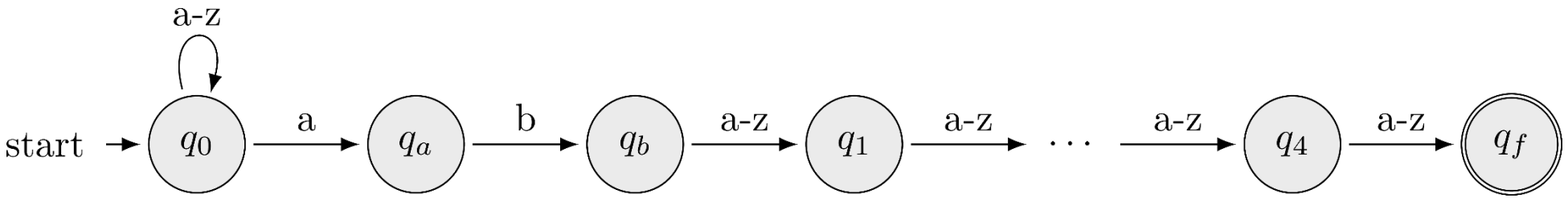
FORMAL LANGUAGES

LANGUAGE	CORRESPONDING AUTOMATON
1. Regular	1. Finite State Machine (DFA, NFA, epsilon-NFA)
2. Context-Free	2. Pushdown Automaton (PDA)
3. Context-Sensitive	3. Linear-Bounded Nondeterministic TM
4. Recursively Enumerable	4. TM

FORMAL LANGUAGES: REGULAR



/a*b/



/.*ab...../

FORMAL LANGUAGES: REGULAR

1. Already useful
2. The corresponding automata are simple (DFA, NFA, with epsilon transitions)
3. They cannot recognize certain strings, e.g. strings where a sequence of “a”s is to be followed by equally many “b”s

FORMAL LANGUAGES: CONTEXT FREE

Definition 1 (Pushdown Automaton (PDA)**.** *A pushdown automaton is an ϵ-NFA with access to a stack, which itself is given by a stack alphabet Γ with initial stack configuration $(Z_0) \in \Gamma$, and such that the transition operation*

$$\delta : Q \times (\{\epsilon\} \cup \Sigma) \times \Gamma \longrightarrow 2^{Q \times \Gamma^*}$$

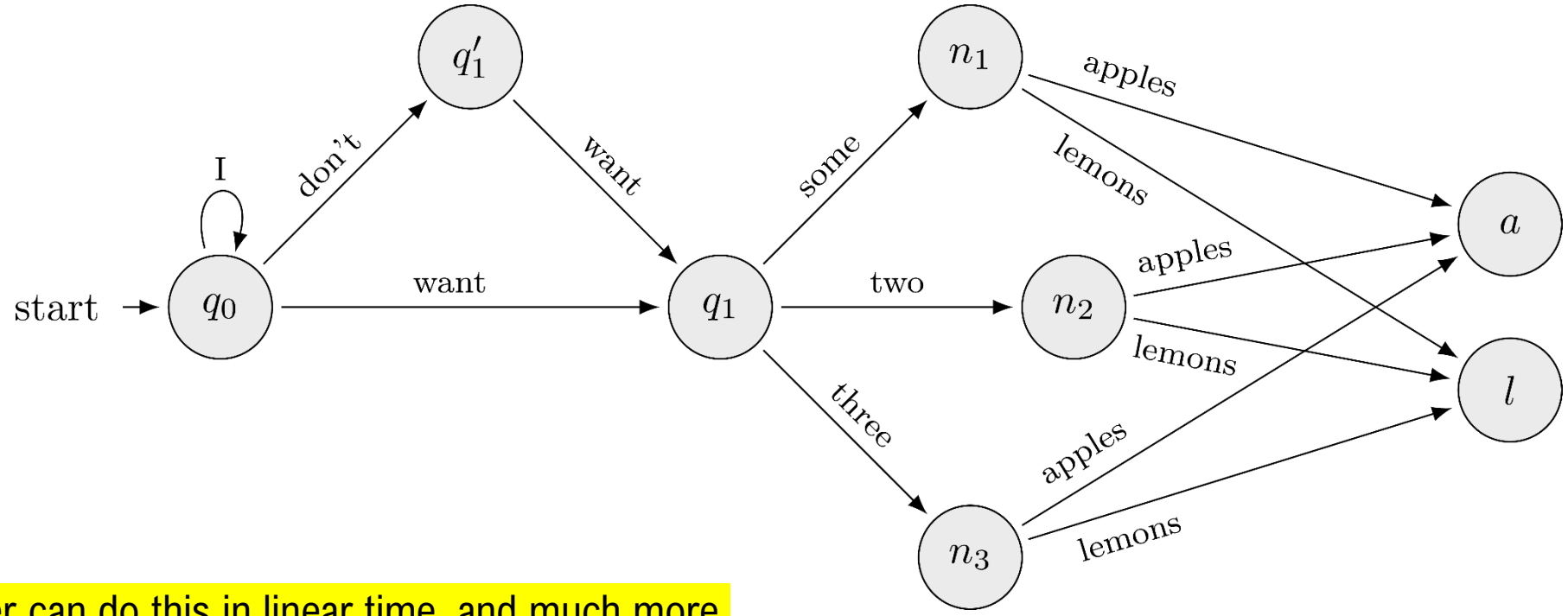
always pops the top element $a \in \Gamma$ off the stack and returns a list of pairs $(q, s) \in Q \times \Gamma^*$ that indicate what state to transition to, and what string to push onto the stack. Here Γ^* is the set of strings of arbitrary length in the stack alphabet, and ϵ is the empty input symbol.

Can store their current state in a limited fashion: much more powerful

FORMAL LANGUAGES

1. Regular and context free languages already capture quite useful classes; many programming languages (e.g. whitespace, but not C++)
2. The English language is not a CFG
3. But: using e.g. an Earley parser, one can decide membership of a string for a given CFG or regular grammar in cubic time in the input length!

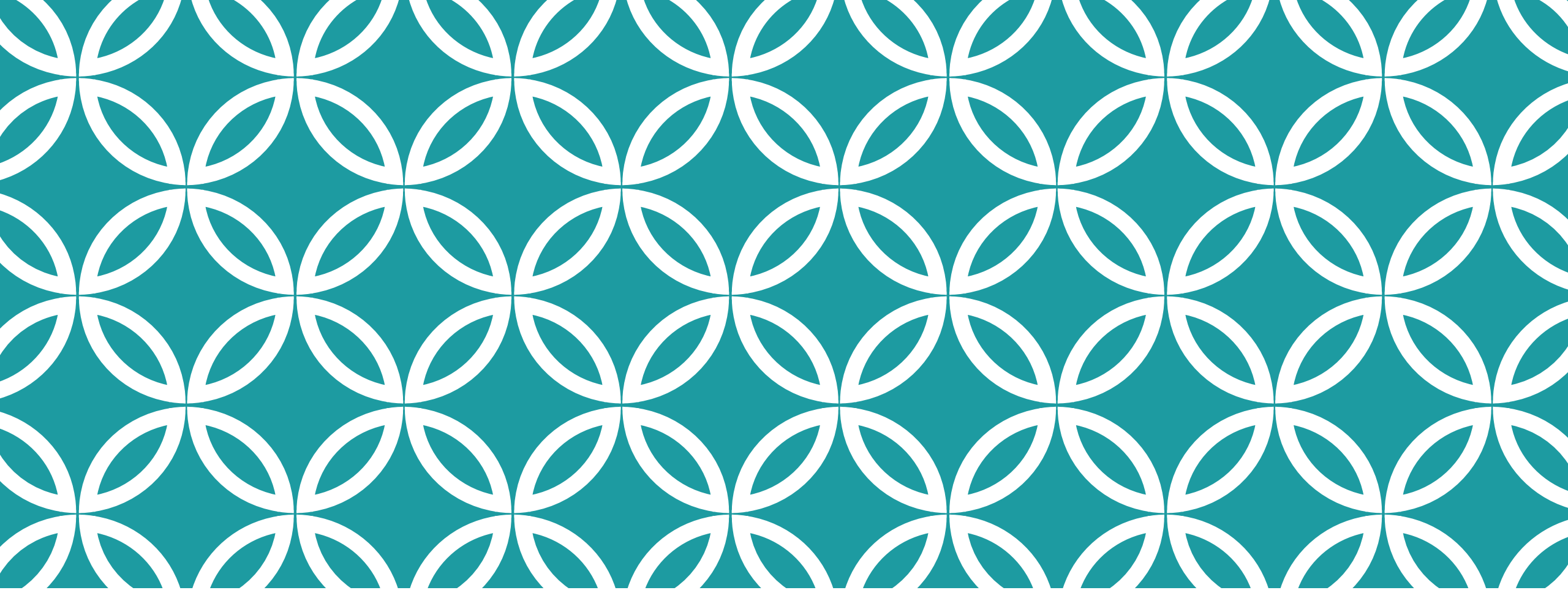
FORMAL LANGUAGES: DIALOGUES?



An Earley parser can do this in linear time, and much more complicated dialogue trees (think: Siri) in cubic time.

A QUANTUM EARLEY PARSER

Remark 1. *There exists a reversible classical algorithm parsing a context free grammar with input size n in time $\mathrm{O}(n^{4.76})$ and space $\mathrm{O}(n^2 \log n)$.*

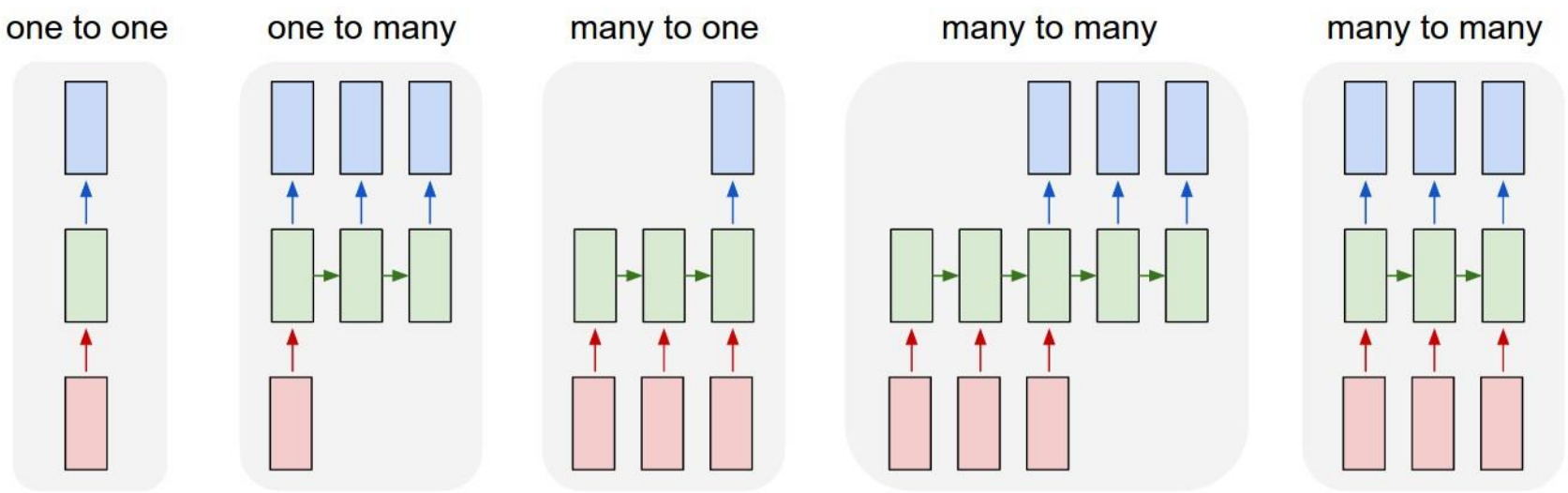


GENERATIVE MODELS

Sequence to sequence
DeepSpeech

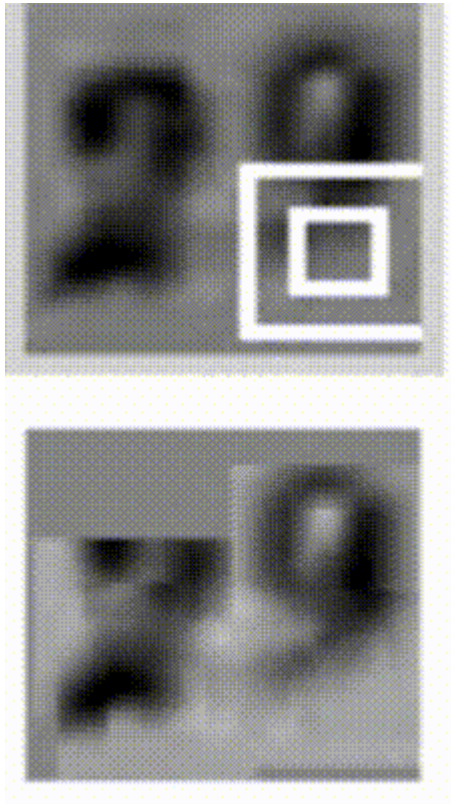
GENERATIVE MODELS

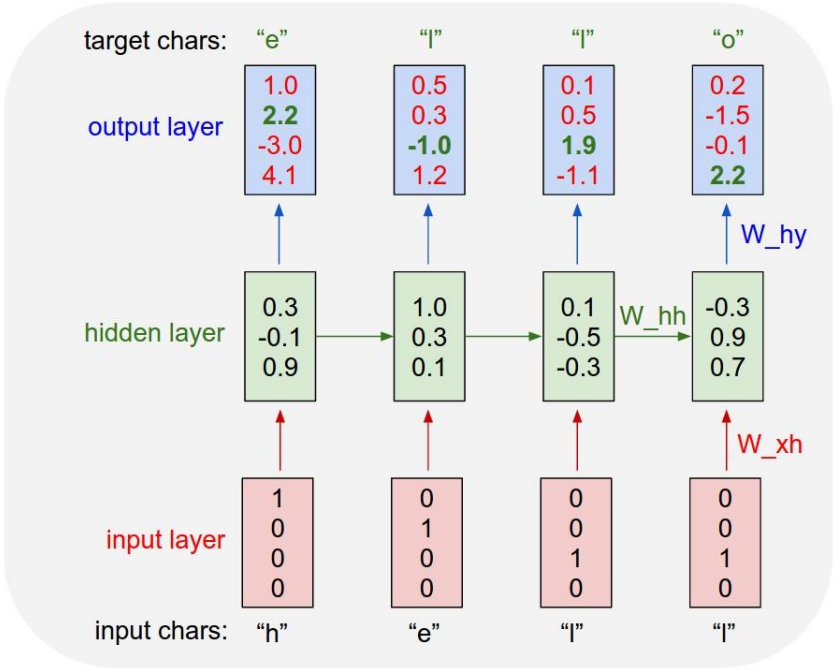
(CREDITS: KARPATHY)

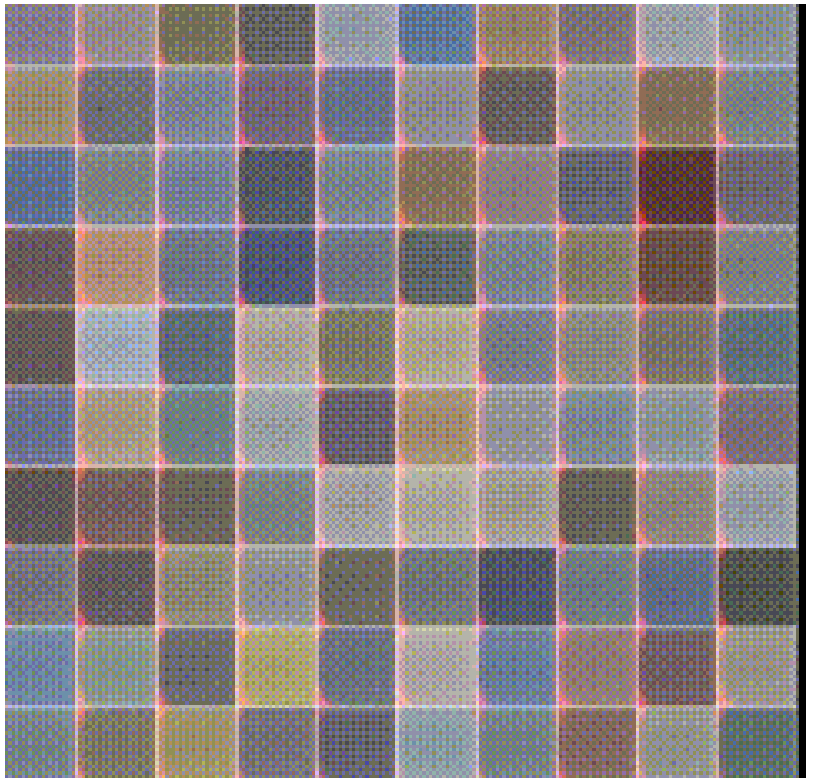


GENERATIVE MODELS

(CREDITS: KARPATHY)





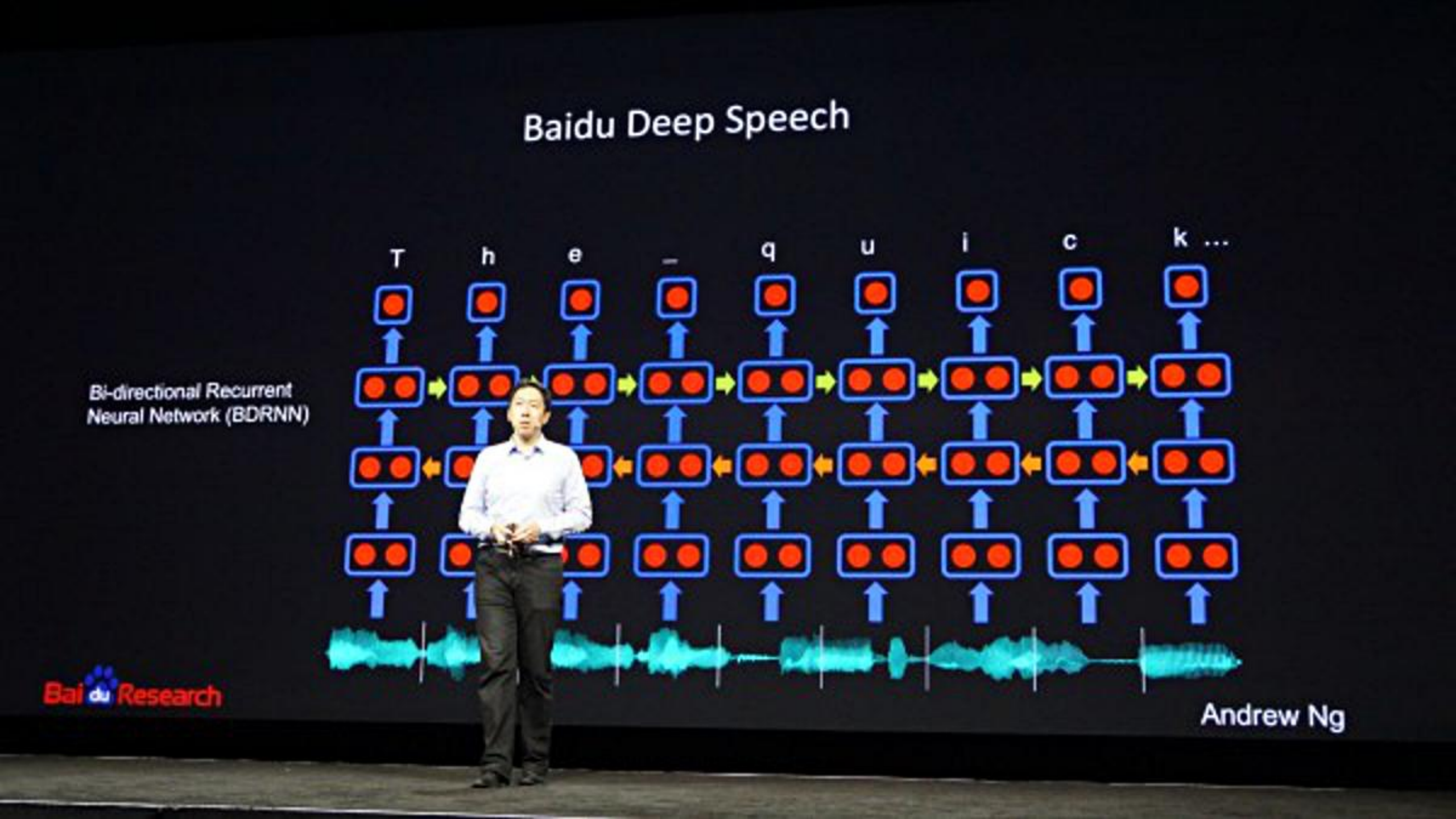


ISN'T THAT SIMILAR TO PROBABILISTIC INPUT TO A PARSER?

1. Parsing heuristics
   Hints where to go first in the parse tree
2. Decoding the output of voice recognition models:
   at each time, instead of a specific letter/word with certainty one obtains a distribution over letters/words.
3. Handwriting recognition, video action labelling…

The common denominator is to decode a probabilistic sequence, and find the one that ultimately (i.e. at the end) has the highest score according to some measure.

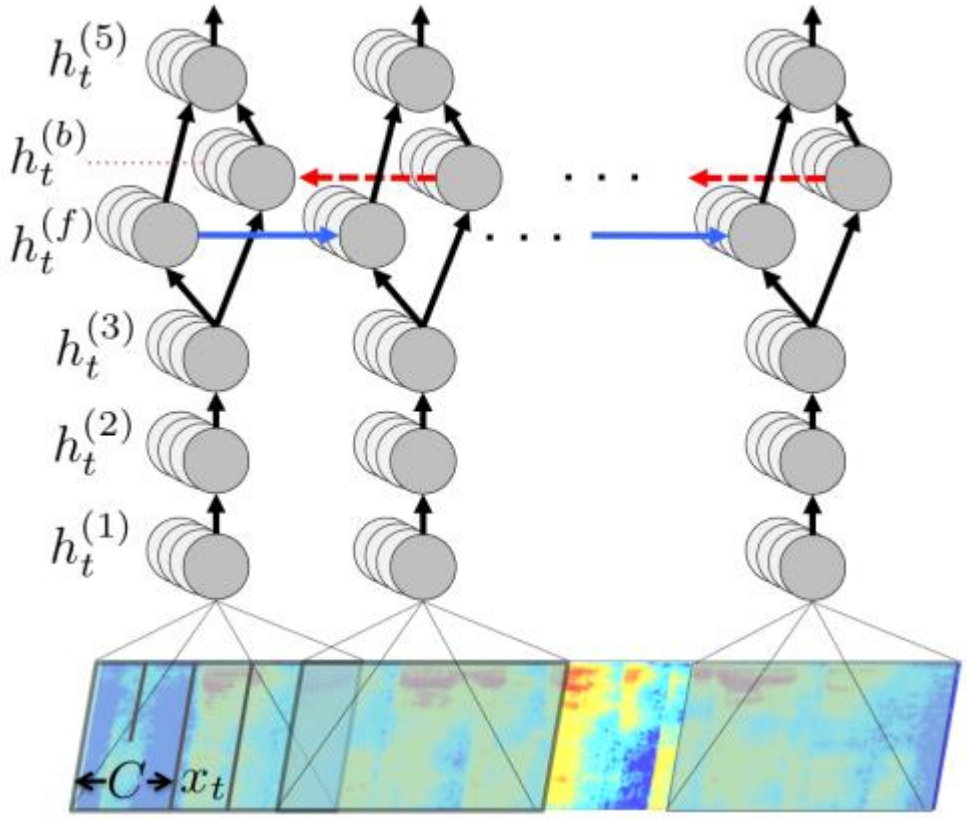
Baidu Deep Speech



DEEPSPEECH

1. State-of-the-art speech recognition system
   BAIDU/MOZILLA ’18/19
   multi-layer LSTM
2. Takes as input mel cepstral coefficients (MFCCs)
   audio frames of 25 ms, 5 ms overlap, DFT, fold with mel scale, log, DCT
3. Trained with CTC Loss on Mozilla’s Common Voice Dataset
4. Output at every frame: letter prediction
   a probability distribution over “a”-”z”, space, and some special markers.

DEEPSPEECH



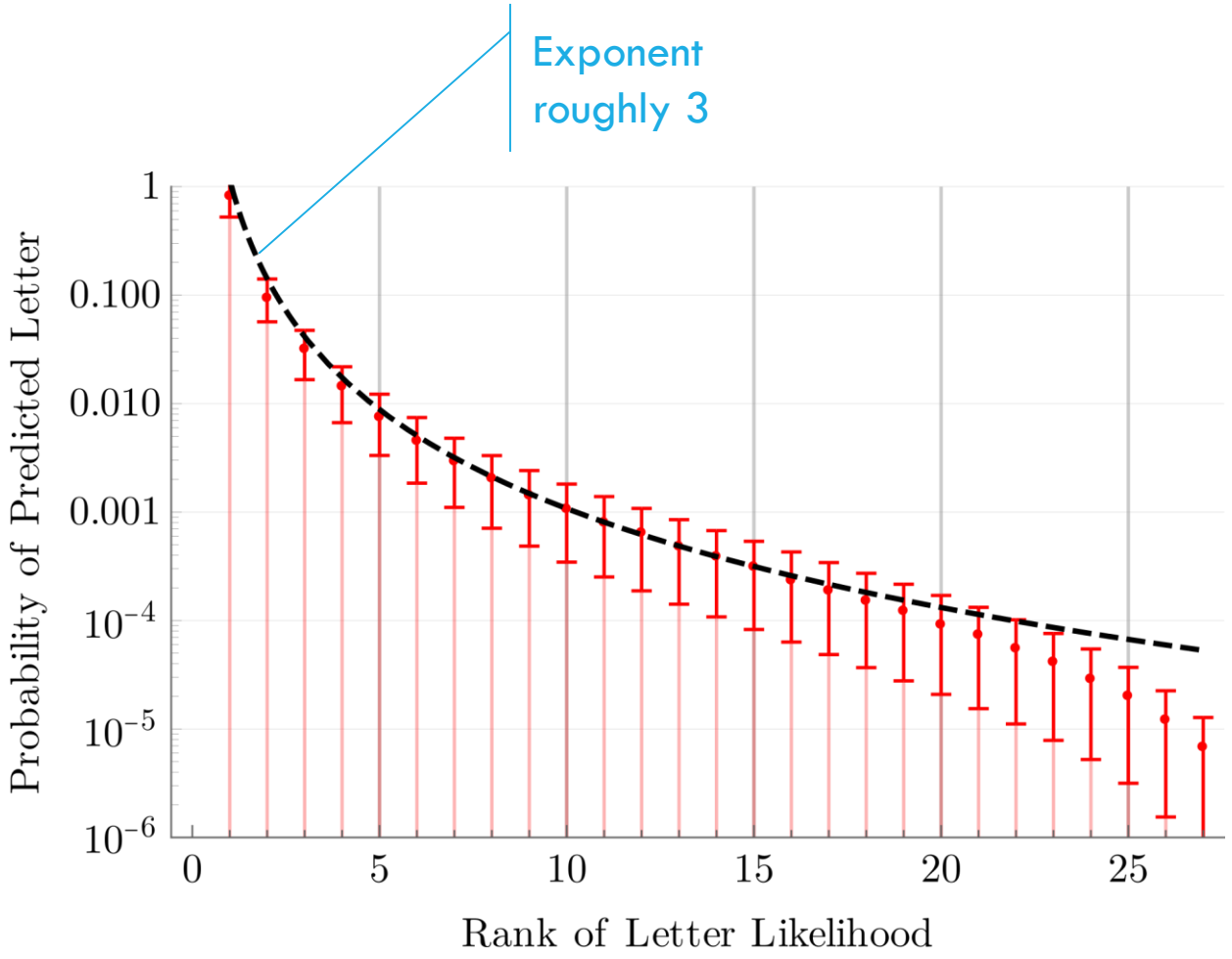
letter	likelihood
A	.03
B	.011
C	.81
D	.05
E	.004
F	.11

⋮

DEEPSPEECH

letter	likelihood
A	.03
B	.011
C	.81
D	.05
E	.004
F	.11
⋮	

rank	likelihood
1	.81
2	.11
3	.05
4	.03
5	.025
6	.011
⋮	



PROBABILISTIC INPUT: KEEP CALM AND CARRY ON WITH THE BEST GUESS

1. This is called greedy decoding, and does not work well.

While hunting in Africa, I shot an elephant in my pajamas. How he got into my pajamas, I don't know.

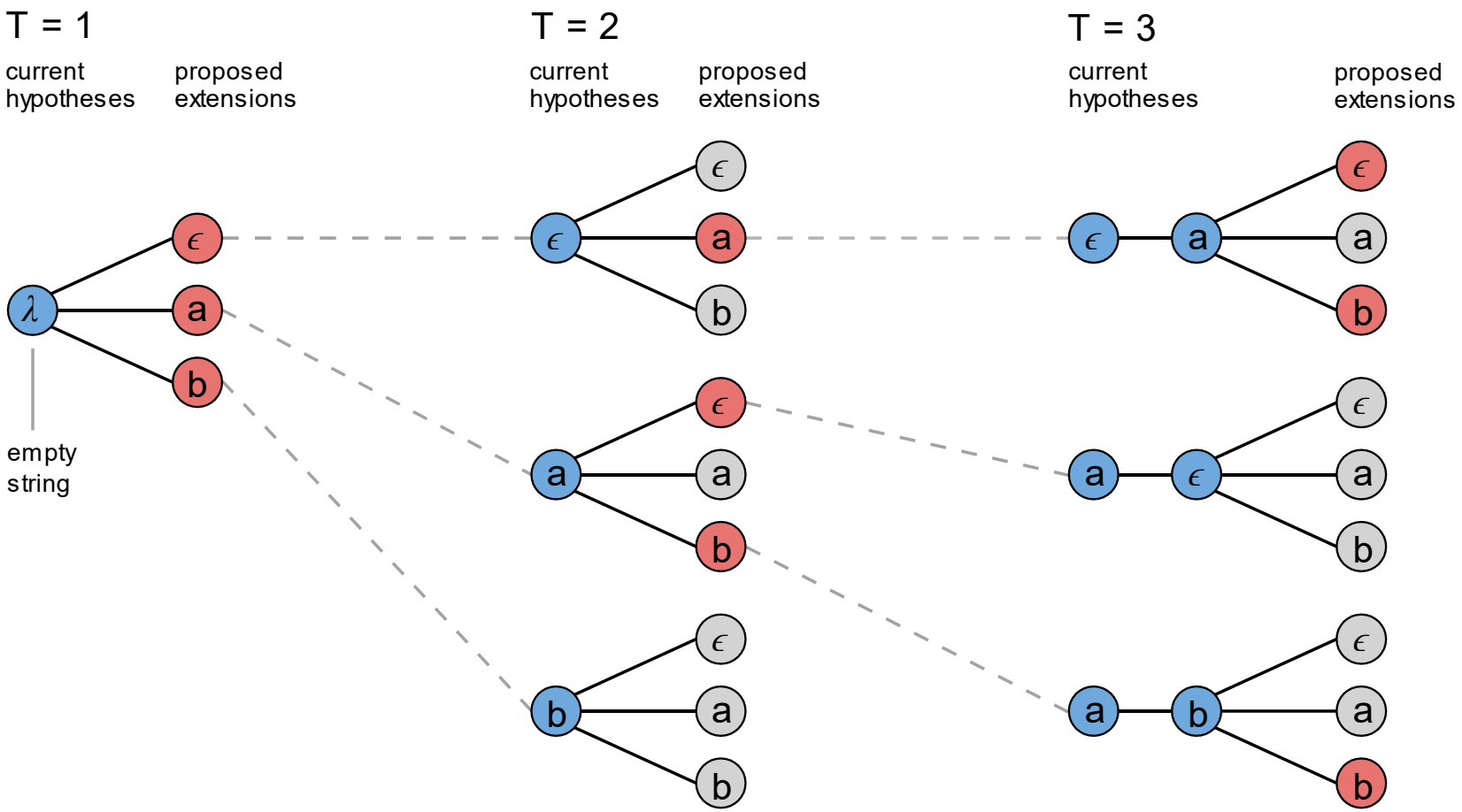
What is the most likely spoken sentence?

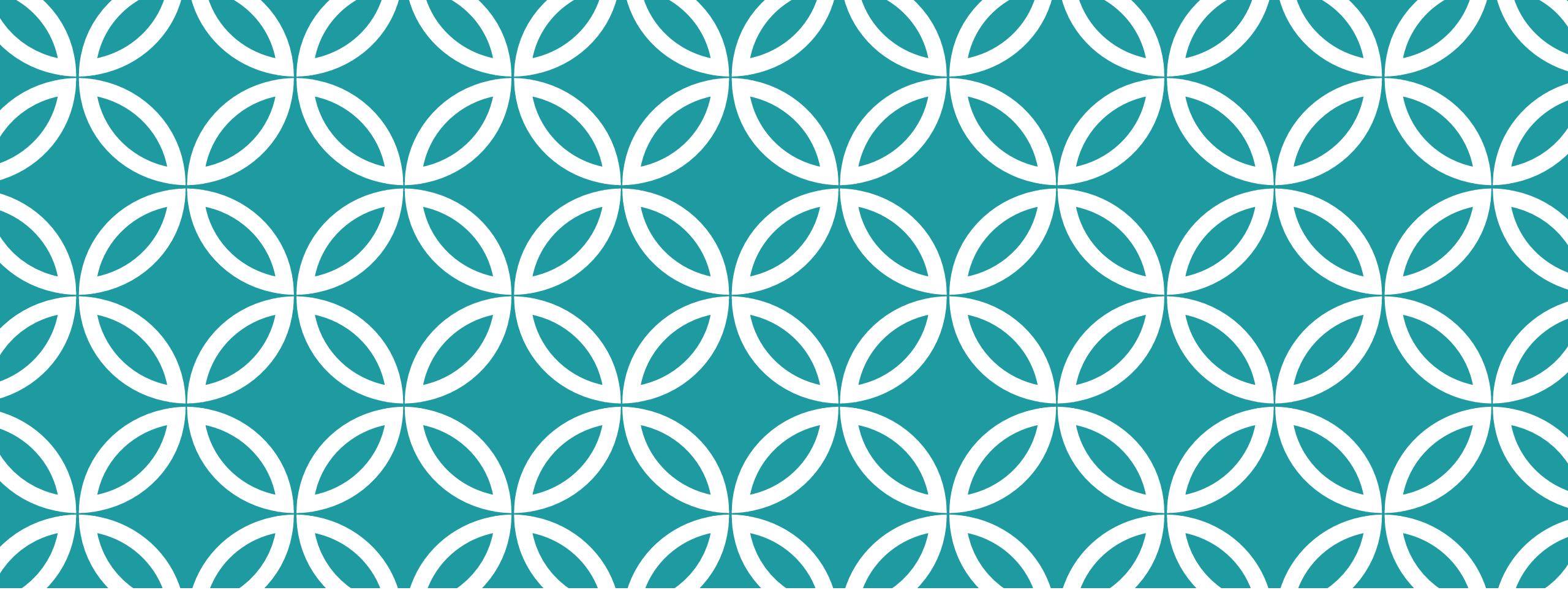
Which one fits into the context best?

2. Keep around multiple options at every point, and toss out bad guesses as we go along.

THIS IS KNOWN AS BEAM SEARCH.

PROBABILISTIC INPUT: BEAM SEARCH





QUANTUM SEARCH DECODER

Formal Problem Statements
Algorithm and Runtime
DeepSpeech Example

FORMAL PROBLEM STATEMENTS

1. What is the most likely parsed path?
2. What is the highest scoring path under a secondary metric?

1 MOST LIKELY PARSE

Input: Decoder M over alphabet Σ and with set of internal configurations Ω. Sequence of random variables $(X_i)_{i \leq n}$ over sample space Σ.

Question: Find $\sigma = \operatorname{argmax}_{x \in \Omega} \Pr(M_n = x)$.

2 HIGHEST SCORE PARSE

Input: Decoder M over alphabet Σ and with state space Ω. Sequence of random variables $(X_i)_{i \leq n}$ over sample space Σ. Scoring function $F : \Omega \longrightarrow \mathbb{R}$. Let the highest-scoring element be $\tau = \operatorname{argmax}_{x \in \Omega} F(x)$.

Promise: $\mathbb{P}(M_n = \text{string with index } i) = p_i = \mathbb{P}(\tau \text{ is at index } i)$.

Question: Find τ.

The input is "good advice" on where to find the best element.

RESULTS

1

MOST LIKELY PARSE

Input: Decoder M over alphabet Σ and with set of internal configurations Ω. Sequence of random variables $(X_i)_{i \leq n}$ over sample space Σ.

Question: Find $\sigma = \text{argmax}_{x \in \Omega} \Pr(M_n = x)$.

Theorem 1. *For an input sequence of length n of random variables to a parser with a classical sampling runtime $T(n)$, there exists a quantum search algorithm answering MOST LIKELY PARSE with certainty, using $\pi/4\sqrt{\Pr(M_n = \sigma)}$ iterations. In each iteration, it runs a quantum circuit for the sampler in $O(T(n)^{1.6})$ time.*

RESULTS

2

HIGHEST SCORE PARSE

Input: Decoder M over alphabet Σ and with state space Ω. Sequence of random variables $(X_i)_{i \leq n}$ over sample space Σ. Scoring function $F : \Omega \longrightarrow \mathbb{R}$. Let the highest-scoring element be $\tau = \text{argmax}_{x \in \Omega} F(x)$.

Promise: $\mathbb{P}(M_n = \text{string with index } i) = p_i = \mathbb{P}(\tau \text{ is at index } i)$.

Question: Find τ.

Theorem 3. *With the same setup as in theorem 1 but under the promise that the input tokens are iid with $X_i \sim \text{Power}_{|\Sigma|}(k)$ over alphabet Σ (definition 8), that the decoder has a branching ratio $R \leq |\Sigma|$, and that we can uniformly sample from the grammar to be decoded, there exists a quantum algorithm QUANTUMSEARCHDECODE answering HIGHEST SCORE PARSE with an expected number of iterations*

$$\text{RT}_1(R, k, n) = \text{O}\left(R^{nf(R,k)}\right), \quad \text{where } f(R, k) = \log\left(\frac{H_R(k/2)}{H_R(k)^{1/2}}\right) \Big/ \log R,$$

and where $H_R(k)$ denotes the R^{th} harmonic number of order k. Each iteration runs a quantum circuit for the sampler in time $\text{O}(T(n)^{1.6})$.

There exists no classical algorithm to solve this problem based on taking stochastic samples from the decoder M that requires less than $\Omega(R^n)$ samples.

ONE STEP BACK:
POWER LAW INPUT = POWER LAW SEARCH SPACE?

– Matt F. May 22 at 15:41

2 If you perform a Fourier expansion of the indicator function $1_{[b',c']}$ and use Fubini's theorem (which requires some preliminary smoothing of the indicator function to justify properly, but never mind that) you can convert the n-dimensional z-integral in the previous comment to a one-dimensional integral over the Fourier variable, which should be a suitable form for instance for working out asymptotics in various limiting regimes such as $n \to \infty$, if that is your application of interest. – Terry Tao May 22 at 18:41

1 Also, these integrals may obey delay-differential equations similar to that obeyed by the Dickman or Buchstab functions. See for instance Exercise 39 of my lecture notes terrytao.wordpress.com/2014/11/23/… – Terry Tao May 22 at 18:48

Terry Tao's suggestion of taking $n \to \infty$ is reasonable...but I have downvoted because

ALGORITHM FOR FULL QUANTUM SEARCH

function QUANTUMSEARCHDECODE$_m(\mathbf{U}_\mu, F)$
  $bestScore \leftarrow -\infty$
  $counter \leftarrow 0$
  repeat
    $cmp \leftarrow (\cdot) \mapsto bestScore < \cdot$ ▷ comparator against current best score
    $|\psi\rangle \leftarrow$ EXPONENTIALSEARCH$(\mathbf{U}_\mu, cmp \circ F)$ ▷ amplify elements larger than pivot
    $bestScore \leftarrow \mathbf{M}_{\text{score}} |\psi\rangle$ ▷ measure new best score
    $counter \leftarrow counter + 1$
  until $counter = m$
end function

ALGORITHM

Algorithm 2 Algorithm for beam search decoding.

function QUANTUMBEAMDECODE$_m(\mathbf{U}_\mu, F, p_0)$

$bestScore \leftarrow -\infty$

$counter \leftarrow 0$

repeat

$cmp_1 \leftarrow [(\cdot) \mapsto (p_0 < \cdot)]$ ▷ comparator against threshold

$cmp_2 \leftarrow [(\cdot) \mapsto (bestScore < \cdot)]$ ▷ comparator against current best score

$amp \leftarrow [(\cdot) \mapsto \text{AMPLITUDEAMPLIFICATION}(\cdot, cmp_1)]$ ▷ prune hypotheses

$|\psi\rangle \leftarrow \text{EXPONENTIALSEARCH}(amp \circ \mathbf{U}_\mu, cmp_2 \circ F)$ ▷ select elements ≥ pivot

$bestScore \leftarrow \mathbf{M}_{\text{score}} |\psi\rangle$ ▷ measure new best score

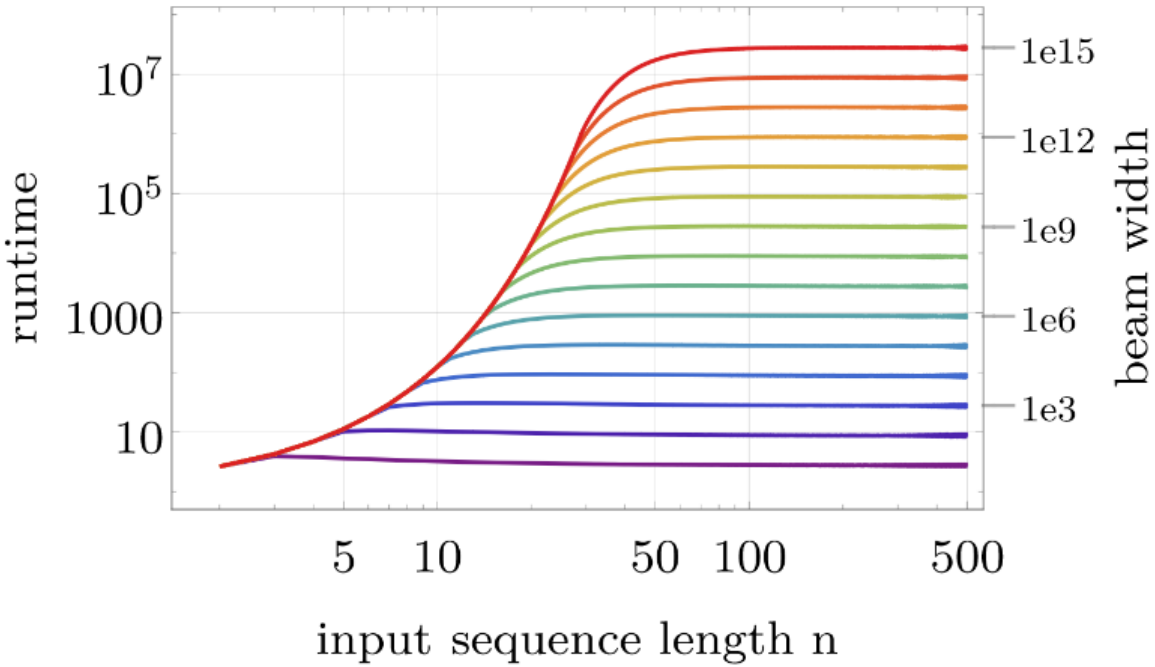
$counter \leftarrow counter + 1$

until $counter = m$

end function

WHAT LENGTH SEQUENCES COULD YOU PARSE WITH DEEPSPEECH? HOW DOES IT COMPARE?

Classical baseline: approximately 200 (Penn Treebank, WikiText2) KHANDELWAL ET AL. '18



SUMMARY…

1. Quantum Search Decoder with Super Grover-seedup.
2. Quantum Sampling Algorithm for RGs/CFGs
3. Quantum Beam Search with better asymptotics
4. This is motivated on a real-world NN, DeepSpeech.

SUMMARY AND CONCLUSION

1. Depends on a quantum sampling algorithm for CFGs.
   How tight is that bound?
2. A wider analysis of sequence to sequence models.
   Machine translation, time series analysis, action tagging for videos etc.
3. How NISQ-y are we?
   Amplitude amplification is not a near-term quantum algorithm.
   But: memory requirement does not grow with beam width, so there will be a crossing point.

THANKS! QUESTIONS?

arXiv link: http://arxiv.org/abs/1909.05023.