Apple - Newton Handwriting Recognition

Snowbird '96

Larry Yaeger
Apple Computer, Inc.

(in collaboration with...)

Handwriting Recognition Team

Core Team

Larry Yaeger	(ATG)
Brandyn Webb	(Contractor)
Dick Lyon	(ATG)
Bill Stafford	(ATG)
Les Vogel	(Contractor)

Other Contributors

Rus Maxham
Gene Ciccarelli
Dan Azuma
Chris Hamlin
Josh Gold
Ernie Beernink
Kara Hayes
Stuart Crawford
Boris Aleksandrovsky
George Mills
Michael Kaplan
Giulia Pagallo

Testers

Polina Fukshansky
Denny Mahdik
Emmanuel Uren
Glen Raphael
Julie Wilson
Ron Dotson

Overview

- **Why, What, and How**
- **Segmentation**
- **Neural Network Issues**
- **Search with Context**
- **Future Directions**

Why Handwriting Recognition?

- Vertical Markets
  - Insurance
  - Hospitals
  - Shipping
  - Copy-Editin~~q~~g

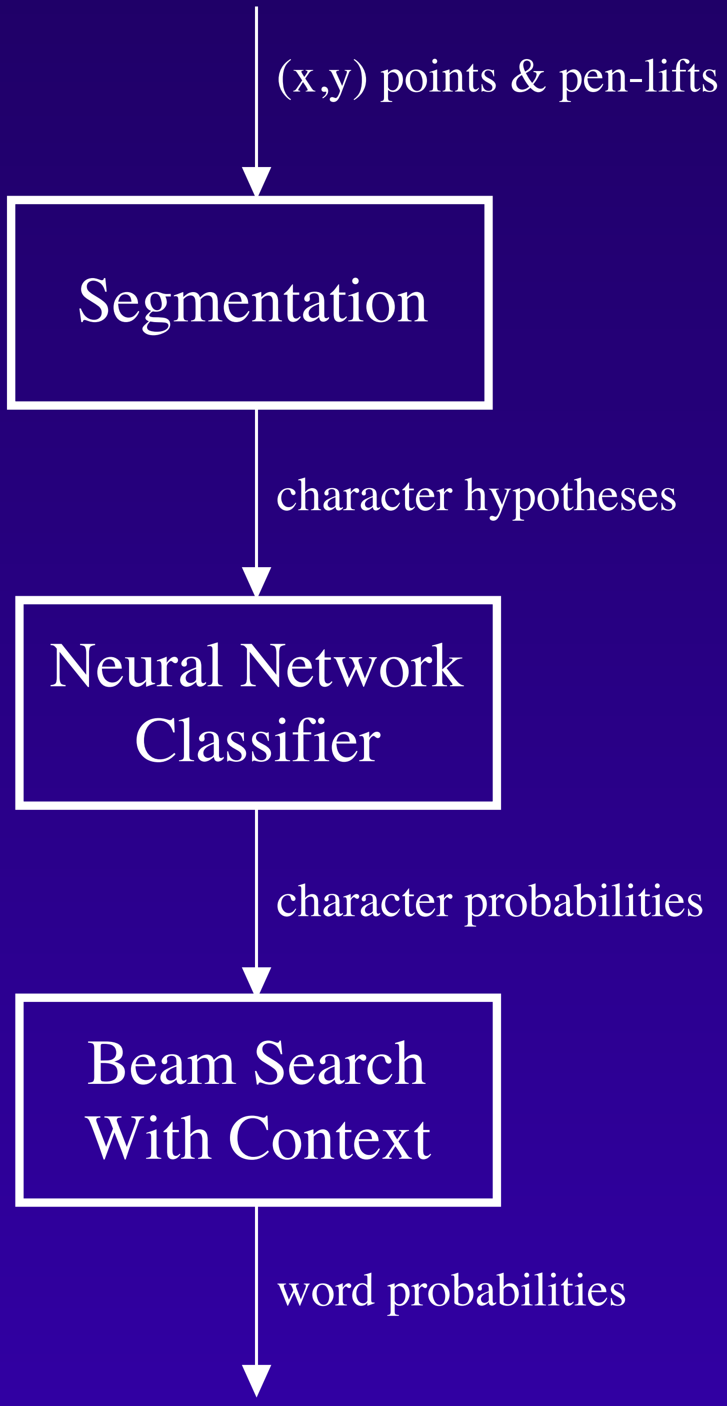
Form-Filling

- Horizontal Markets
  - Non-Typists & Computerphobes
    - “If it doesn't have a keyboard, it's not a computer”
  - PDA's & True Notebook Computers

- Foreign Markets
  - Ideographic languages

New Markets

ANHR's Pipeline Architecture



(x,y) points & pen-lifts

Segmentation

character hypotheses

Neural Network Classifier

character probabilities

Beam Search With Context

word probabilities

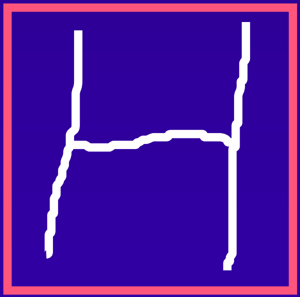
Integrated Segmentation and Recognition

- Which Strokes Comprise Which Characters?

- Constraints
  - All Strokes Must Be Used
  - No Strokes May Be Used Twice

- Efficient Presegmentation
  - Avoid Trying All Possible Permutations
  - Based on Overlap, Crossings, Aspect Ratio, etc.

- Full Printable ASCII Presents Some Challenges



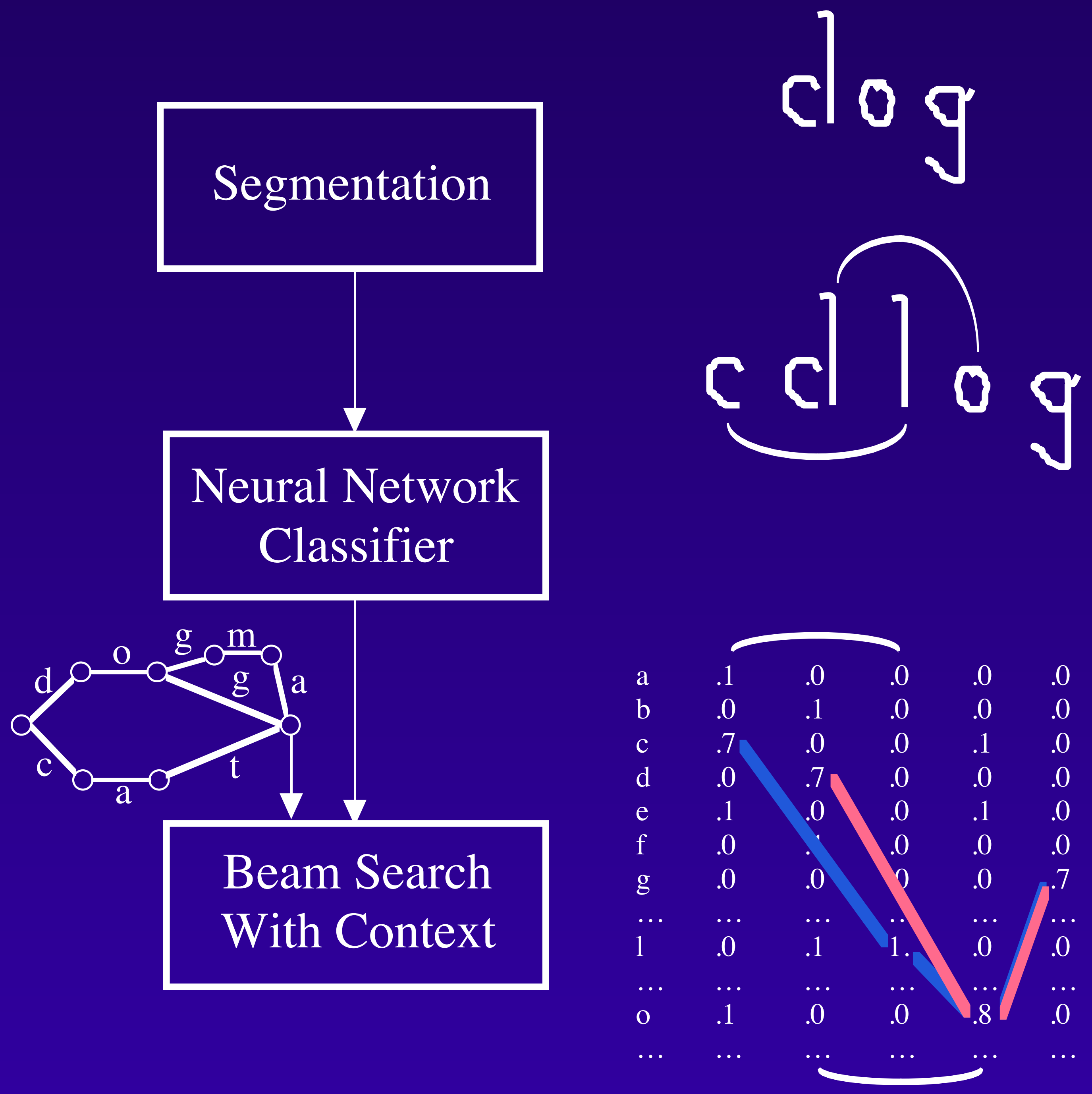
Neural Network Classifier

- Inherently Data-Driven
- Learn from Examples
- Non-Linear Decision Boundaries
- Effective Generalization

Context Is Essential

- Humans Achieve 90% Accuracy on Characters in Isolation (for Our Database)
  - Word Accuracy Would Then Be ~ 60% or Less (.9^5)
- Variety of Context Models Are Possible
  - N-Grams
  - Word Lists
  - Regular Expression Graphs
- "Out of Context" Models Also Necessary
  - "xyzzy", Unix Pathnames, Technical/Medical Terms, etc.

ANHR's Pipeline Architecture



Segmentation

Neural Network Classifier

Beam Search With Context

a	.1	.0	.0	.0	.0
b	.0	.1	.0	.0	.0
c	.7	.0	.0	.1	.0
d	.0	.7	.0	.0	.0
e	.1	.0	.0	.1	.0
f	.0	.1	.0	.0	.0
g	.0	.0	.0	.0	.7
…	…	…	…	…	…
l	.0	.1	1.	.0	.0
…	…	…	…	…	…
o	.1	.0	.0	.8	.0
…	…	…	…	…	…

Segmentation

Segmentation

Ink	Segment Number	Segment	Stroke Count	Forward Delay	Reverse Delay
clog	1	c	1	3	1
	2	cl	2	4	2
	3	clo	3	4	3
	4	l	1	2	1
	5	lo	2	2	2
	6	o	1	1	1
	7	g	1	0	1

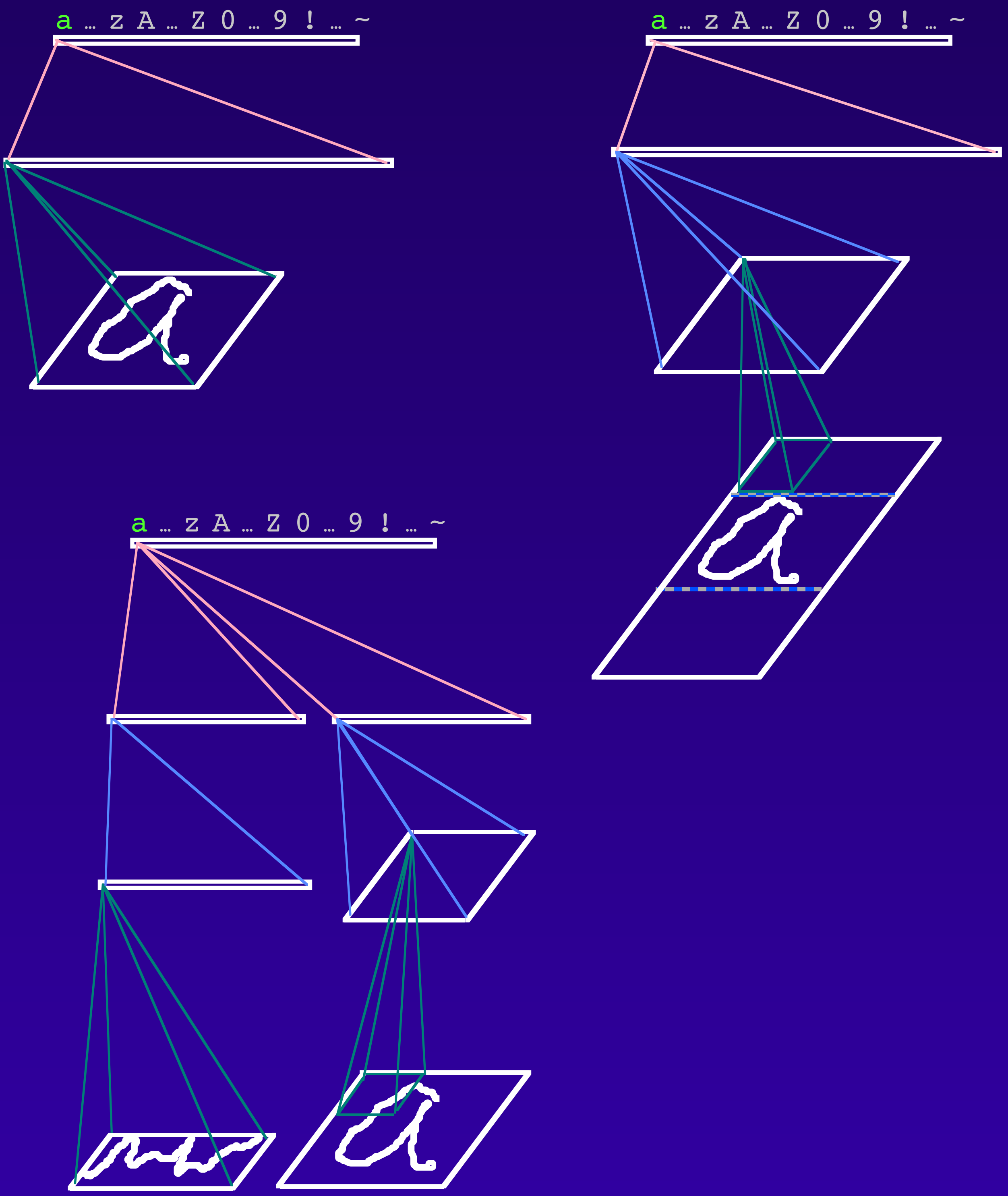
Neural Network Classifier

Network Design

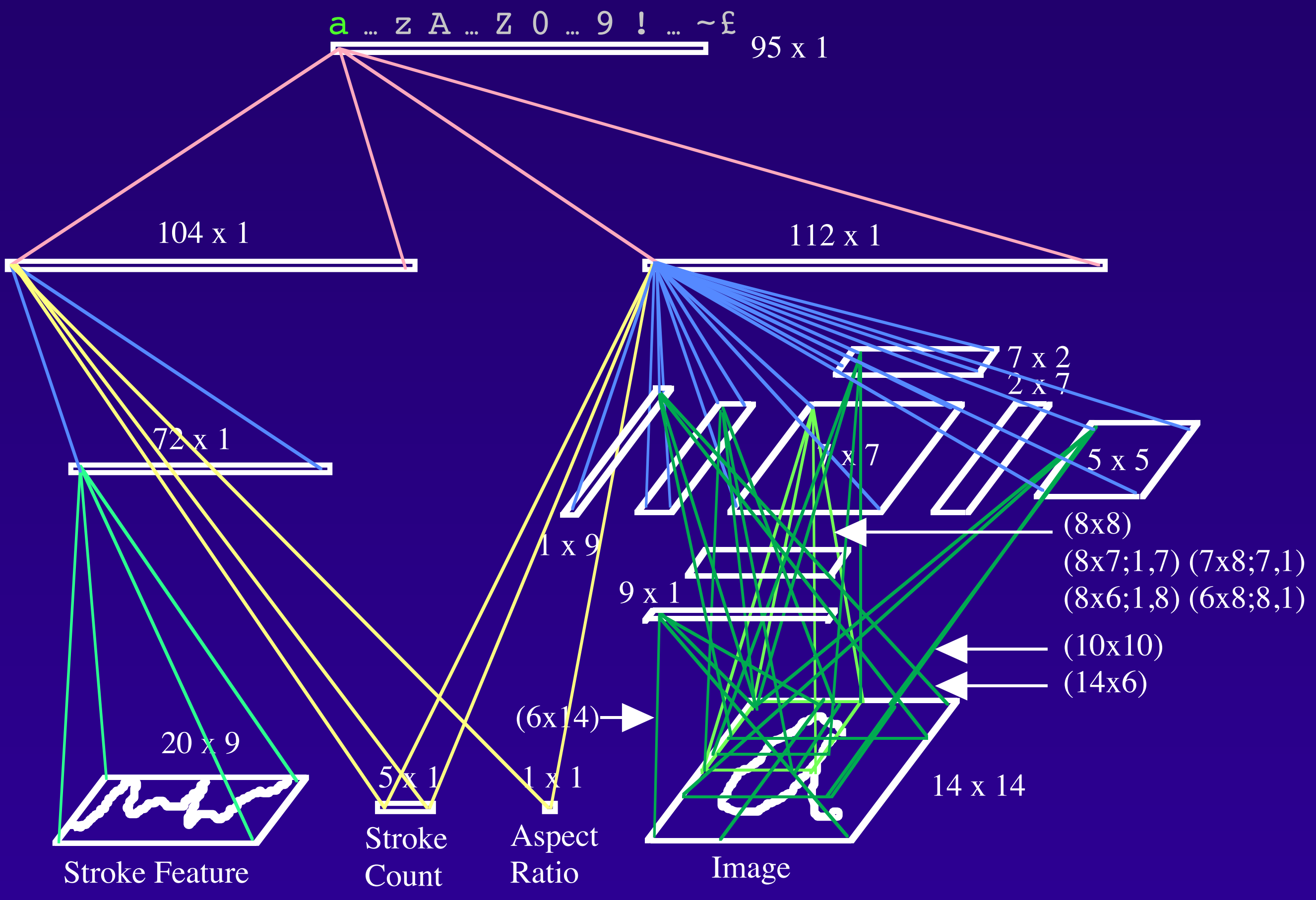
- Variety of Architectures Tried
  - Single Hidden Layer, Fully-Connected
  - Multi-Hidden Layer, Receptive Fields
  - Parallel Classifiers Combined at Output Layer

- Representation as Important as Architecture
  - Anti-Aliased Images
  - Baseline-Driven with Ascenders and Descenders
  - Stroke-Features

Network Architectures



Network Architecture



ATG

Handwriting Recognition

Normalized Output Error

- Based on Recognition of Fact that Most Training Signals are Zero
    - Training Vector for Letter "x"

a	…	w	x	y	z	A	…	Z	0	…	9	!	…	~
0	…	0	1	0	0	0	…	0	0	…	0	0	…	0

- Forces Net to Attempt to Make Unambiguous Classifications

- Difficult to Obtain Meaningful 2nd and 3rd Choice Probabilities

Normalized Output Error

- We Reduce the BP Error for Non-Target Classes Relative to the Target Class
    - By a Factor that "Normalizes" the Non-Target Error Relative to the Target Error, Based on the Number of Non-Target vs. Target Classes
- For Non-Target Output Nodes

$$e' = e \quad 1 / d \, (N_{outputs} - 1)$$

- Allocates Network Resources to Model Low-Probability Regime

Normalized Output Error

- Converges to MMSE Estimate of

  $$f(P(class|input),A)$$

- We Derived that Function:

  $$\langle \hat{e}^2 \rangle = p\ (1-y)^2 + A\ (1-p)\ y^2$$

  where

  $$p = P(class|input),$$
  $$A = 1\ /\ d\ (N_{outputs} - 1)$$

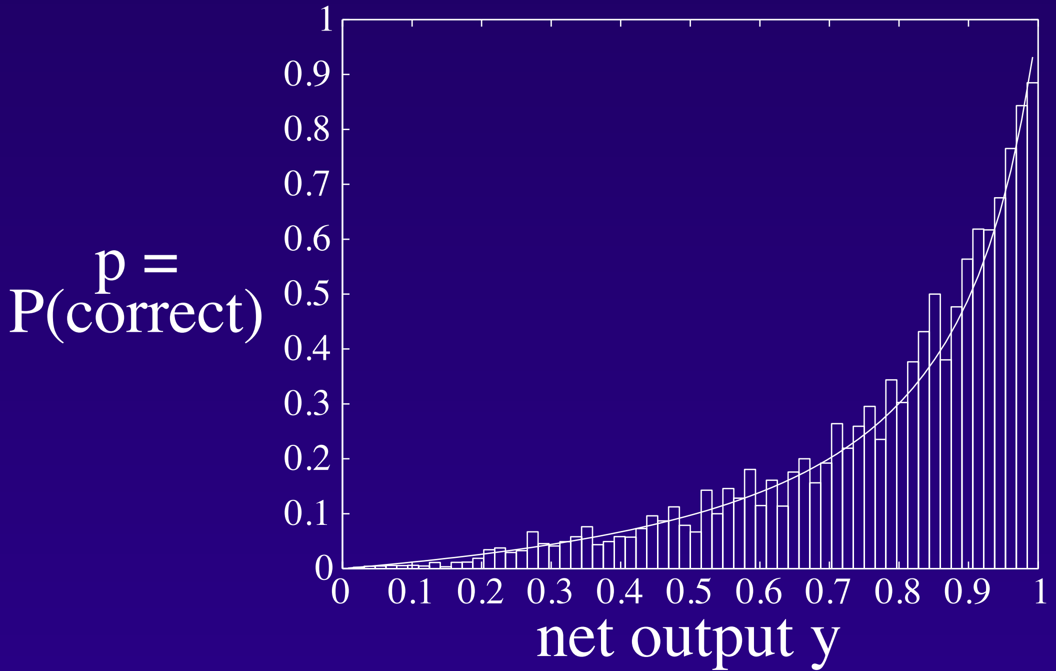
- Output y for Particular Class is Then:

  $$y = p\ /\ (A - A\ p + p)$$

- Inverting for p:

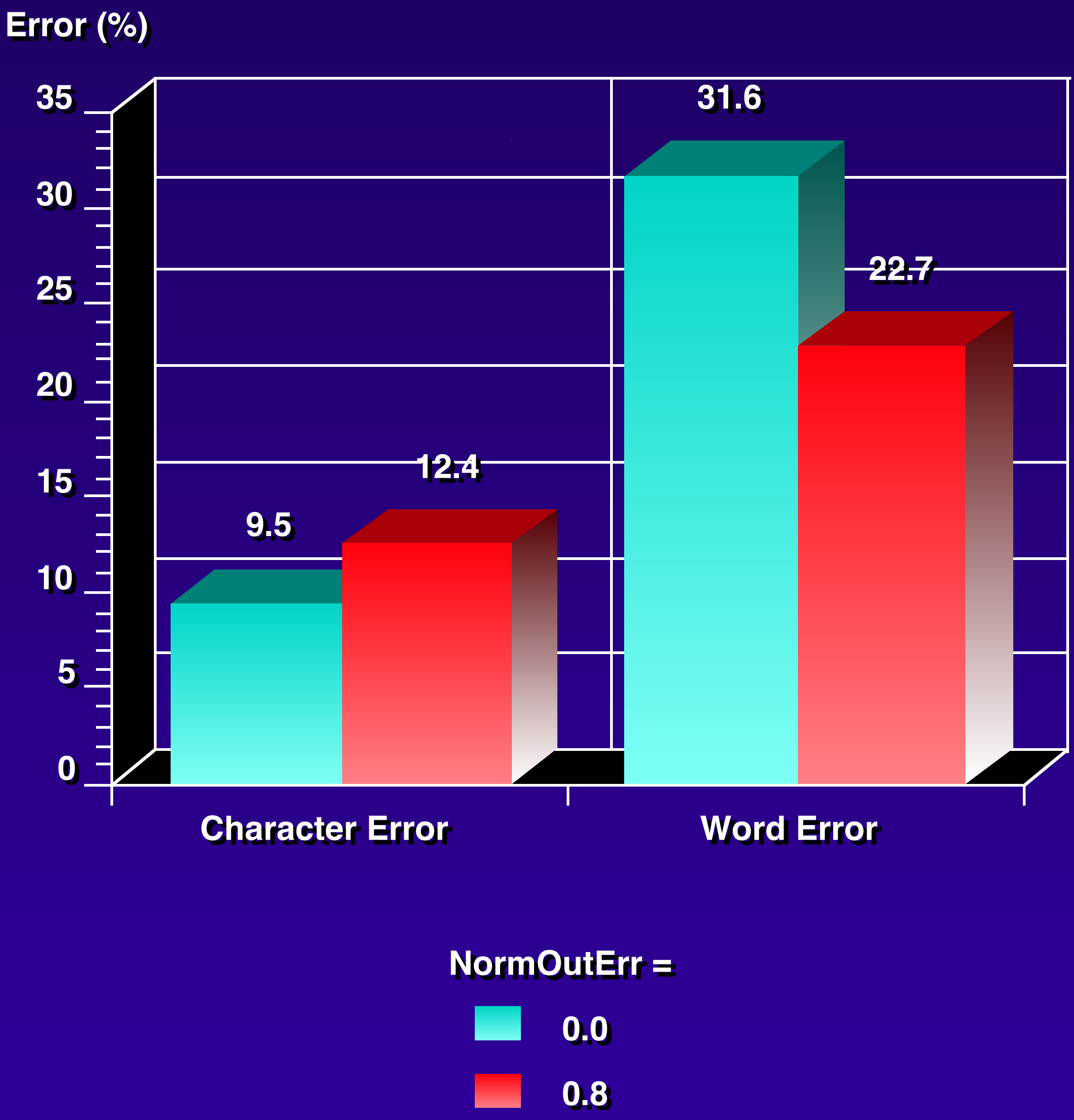
  $$p = y\ A\ /\ (y\ A - y + 1)$$

Normalized Output Error



Empirical p vs. y histogram for a net trained with `A`=0.11 (`d`=0.1), with corresponding theoretical curve

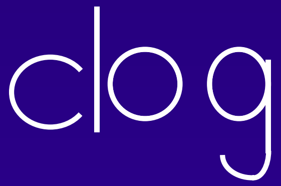
Normalized Output Error



Negative Training

- Inherent Ambiguities Force Segmentation Code to Generate False Segmentations

- Ink Can Be Interpreted in Various Ways...



  - "dog", "clog", "cbg", "%g"

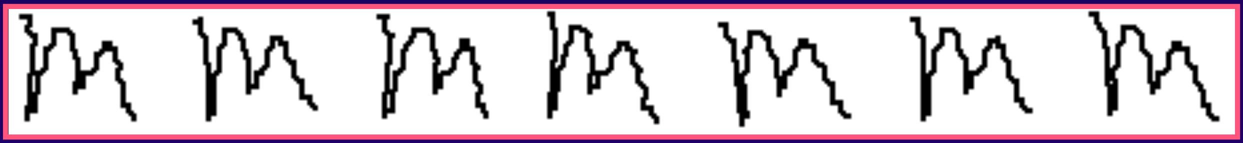
- Train Network to Compute Low Probabilities for False Segmentations

Negative Training

- Modulate Negative Training by
  - Negative Error Factor (0.2 to 0.5)
    - Like `A` in Normalized Output Error
  - Negative Training Probability (0.05 to 0.3)
    - Also Speeds Training
- Too Much Negative Training
  - Suppresses Net Outputs for Characters that Look Like Elements of Multi-Stroke Characters

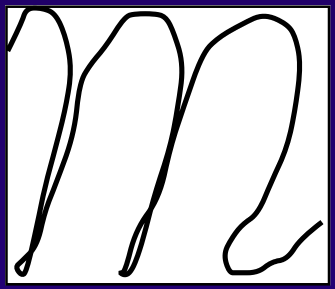
    `(I, 1, l, o, O, 0)`
- Slight Reduction in Character Accuracy, Large Gain in Word Accuracy

Stroke Warping

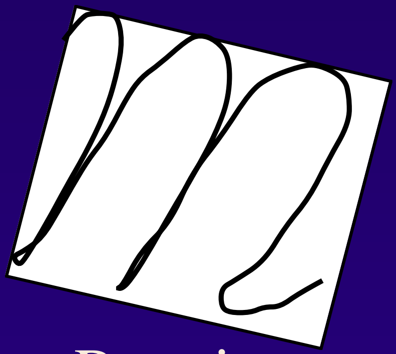


- Produce Random Variations in Stroke Data During Training
- Small Changes in Skew, Rotation, X and Y Linear and Quadratic Scaling
- Consistent with Stylistic Variations
- Improves Generalization by Effectively Adding Extra Data Samples

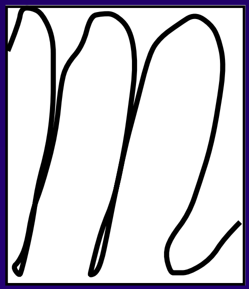
Stroke Warping



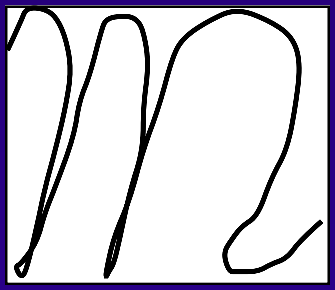
Original



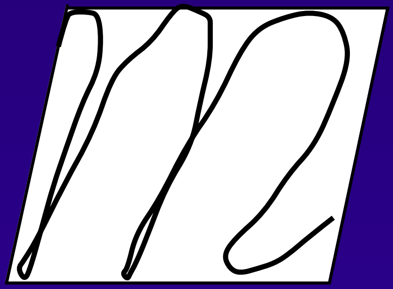
Rotation



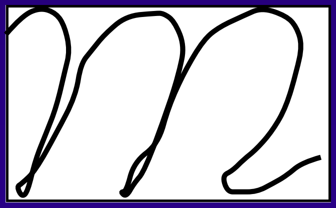
X Linear



X Quadratic



X Skew



Y Linear

Frequency Balancing

- Skip and Repeat Patterns to Balance Class Frequencies
- Instead of Dividing by the Class Priors
  - Produces Noisy Estimate of Low Freq. Classes
  - Requires Renormalization
- Compute Normalized Frequency, Relative to Average Frequency

$$F_i = S_i \; / \; (\; 1/C \sum_{j=1}^{C} S_j \;)$$

Frequency Balancing

- Compute Repetition Factor

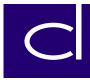
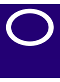
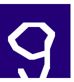
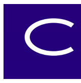
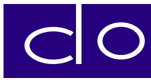
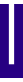
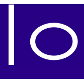
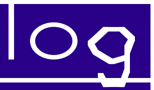
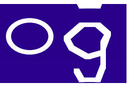
$$R_i = (a / F_i)^b$$

- Where a (0.2 to 0.8) Controls Amount of Skipping vs. Repeating
- And b (0.5 to 0.9) Controls Amount of Balancing

Error Emphasis

- Probabilistically Skip Training for Correctly Classified Patterns
- Never Skip Incorrectly Classified Patterns
- Just One Form of Error Emphasis
  - Can Reduce Learning Rate/Error for Correctly Classified Patterns
  - And Increase Learning Rate/Error for Incorrectly Classified Patterns

Training Probabilities and Error Factors

Segment	Type	Prob. of Usage		Error Factor	
		Correct	Incorrect	Target Class	Other Classes
	POS	0.5	1.0	1.0	0.1
					
					
	NEG	0.18		NA	0.3
					
					
					
					
					

Annealing

- Start with Large Learning Rate, then Decay
  - When Training Set's Total Squared Error Increases
- Start with High Error Emphasis and Frequency Balancing, then Decay

Training Schedule

Phase	Epochs	Learning Rate	Correct Train Prob	Negative Train Prob
1	25	1.0 - 0.5	0.1	0.05
2	25	0.5 - 0.1	0.25	0.1
3	50	0.1 - 0.01	0.5	0.18
4	30	0.01 - 0.001	1.0	0.3

Quantized Weights

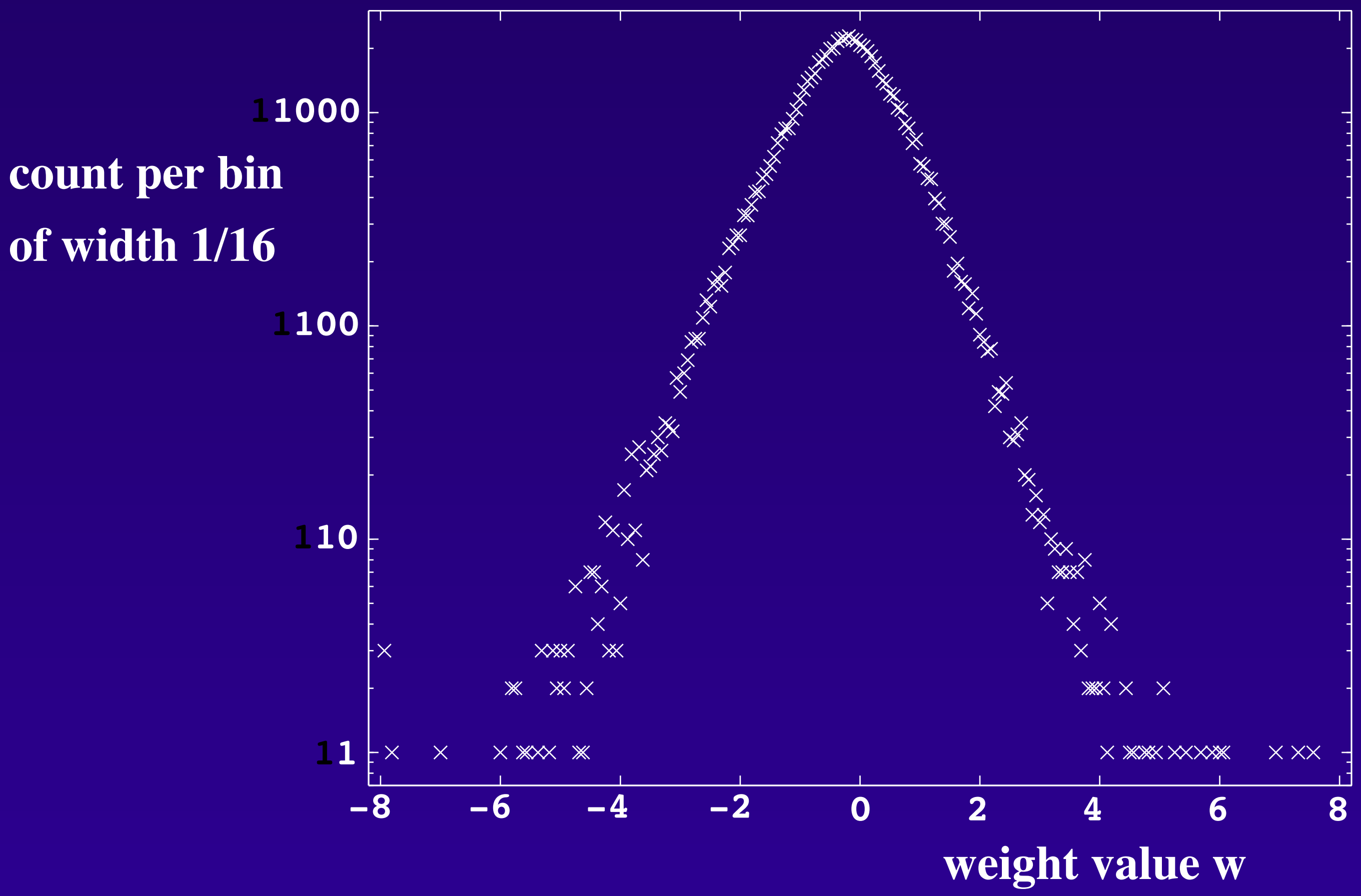
- Forward/Classification Pass Requires Less Precision Than Backward/Learning Pass

- Use One-Byte Weights for Classification
  - Saves Both Space and Time
  - ±3.4 (-8 to +8 with 1/16 Steps)

- Use Three-Byte Weights for Learning
  - ±3.20

- Newton Version Currently
  - ~200KB ROM (~85KB for weights)
  - ~5KB-100KB RAM
  - ~3.8 Char/Second

Quantized Weights

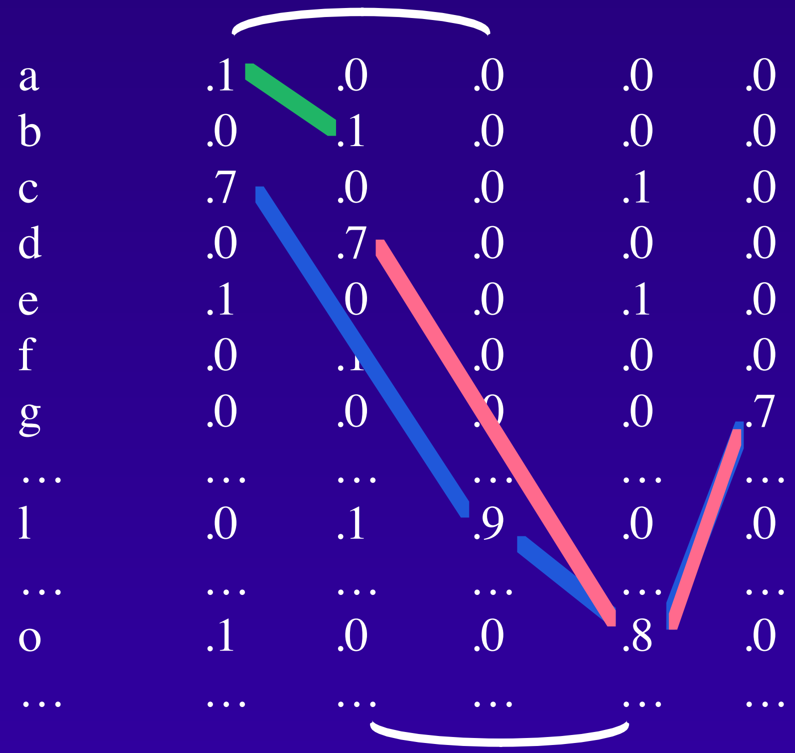


Search with Context

Viterbi Beam Search

- Viterbi: Only One Path Per Node is Required for Global Optimum

- Beam: Low Probability Paths are Unlikely to Overtake Most Likely Paths



a	.1	.0	.0	.0	.0
b	.0	.1	.0	.0	.0
c	.7	.0	.0	.1	.0
d	.0	.7	.0	.0	.0
e	.1	.0	.0	.1	.0
f	.0	.1	.0	.0	.0
g	.0	.0	.0	.0	.7
…	…	…	…	…	…
l	.0	.1	.9	.0	.0
…	…	…	…	…	…
o	.1	.0	.0	.8	.0
…	…	…	…	…	…

Integration with Character Segmentation

- Search Takes Place Over Segmentation Hypotheses (as Well as Character Hypotheses)

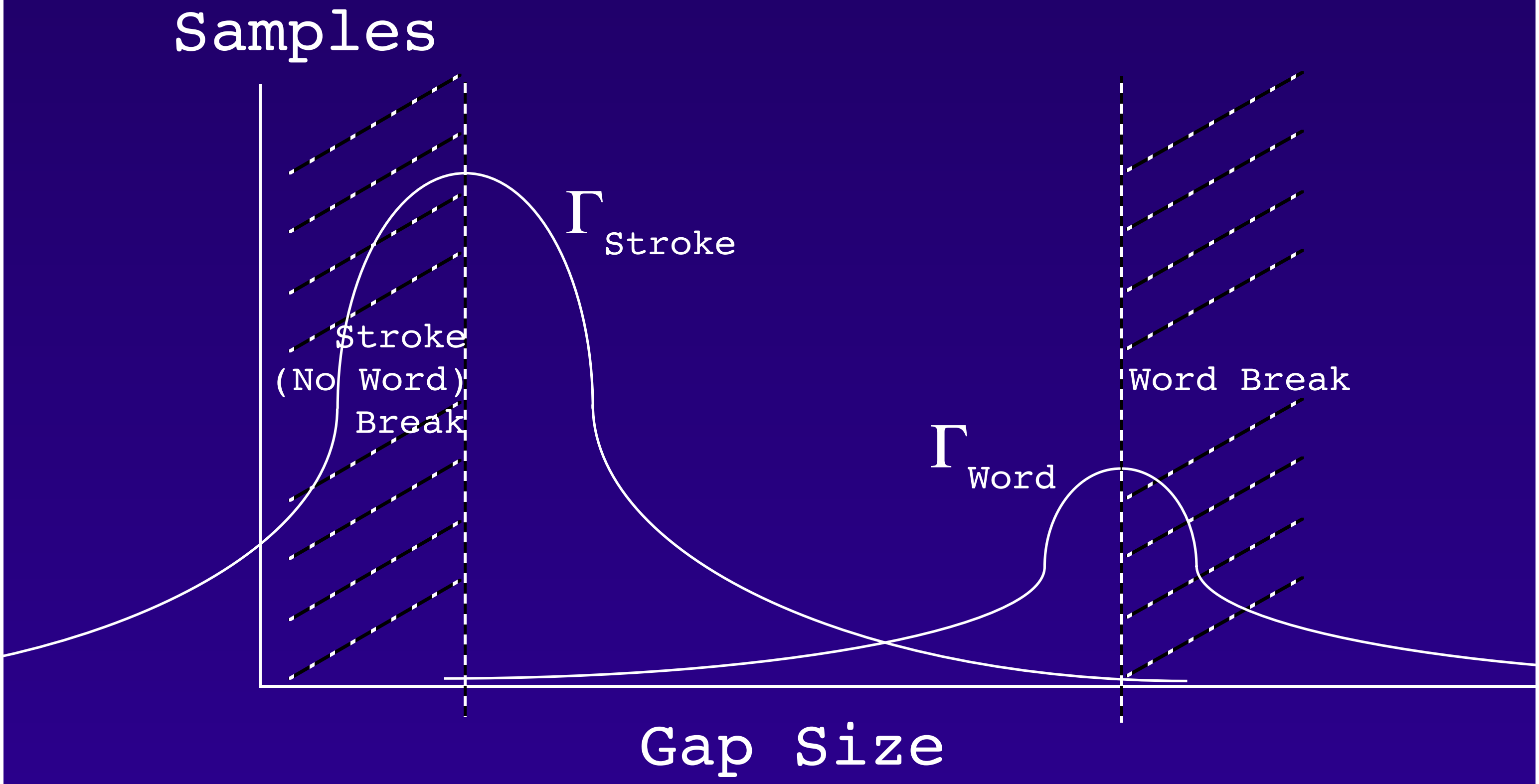
- Stroke Recombinations are Presented in Regular, Predictable Order

- Forward and Reverse "Delay" Parameters Suffice to Indicate Legal Time-Step Transitions

Integration with Word Segmentation

- Search Also Takes Place Over Word Segmentation Hypotheses
- Word-Space Becomes an Optional Segment/Character
  - Weighted by Probability ("SpaceProb") Derived from Statistical Model of Gap Sizes and Stroke Centroid Spacing
- Non-Space Hypothesis is Weighted by 1-SpaceProb

Word Segmentation Statistical Model



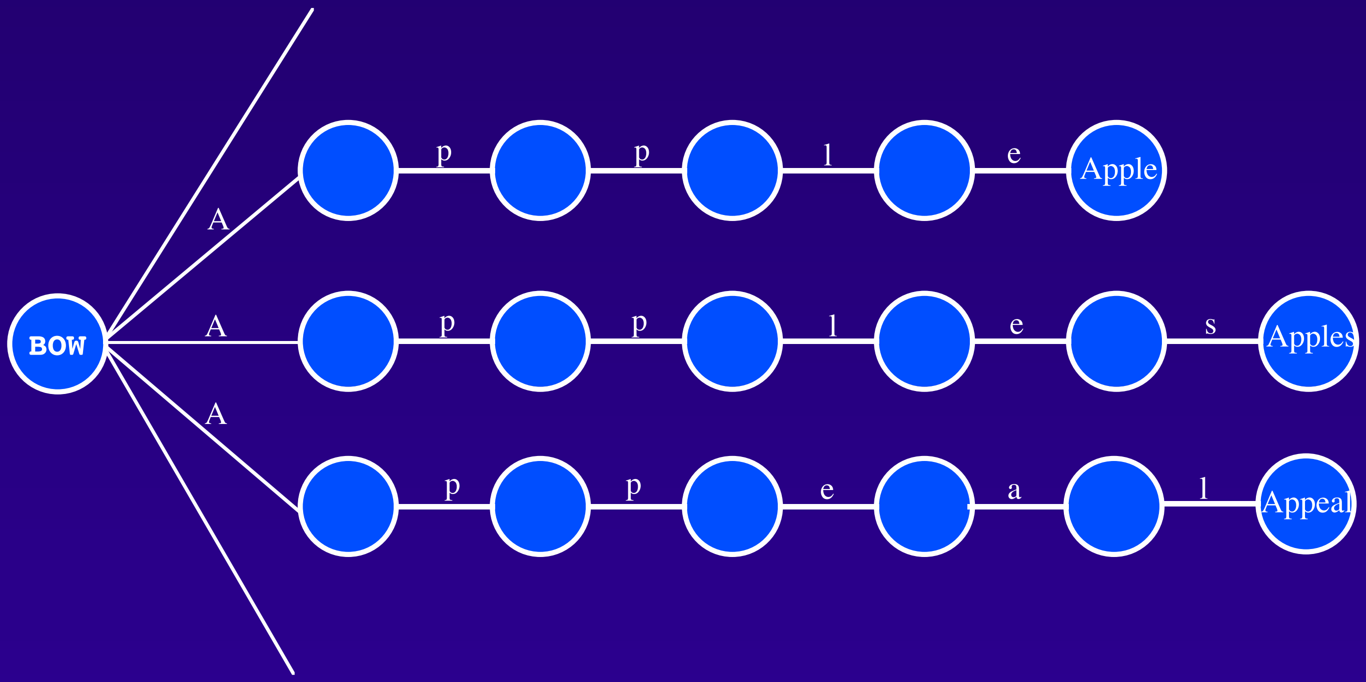
$$P_{\text{Word}} = \Gamma_{\text{Word}} / (\Gamma_{\text{Stroke}} + \Gamma_{\text{Word}})$$

Integration with Context

- Lexical Context Graphs Guide Search
- Each Graph May or May Not Have Letter Transition Probabilities
  - "Langs" Do
  - "Dicts" Do Not
- Langs and Dicts Are Created from
  - Word Lists
  - Regular Expression Grammar
- Multiple Langs and Dicts Are Searched Simultaneously

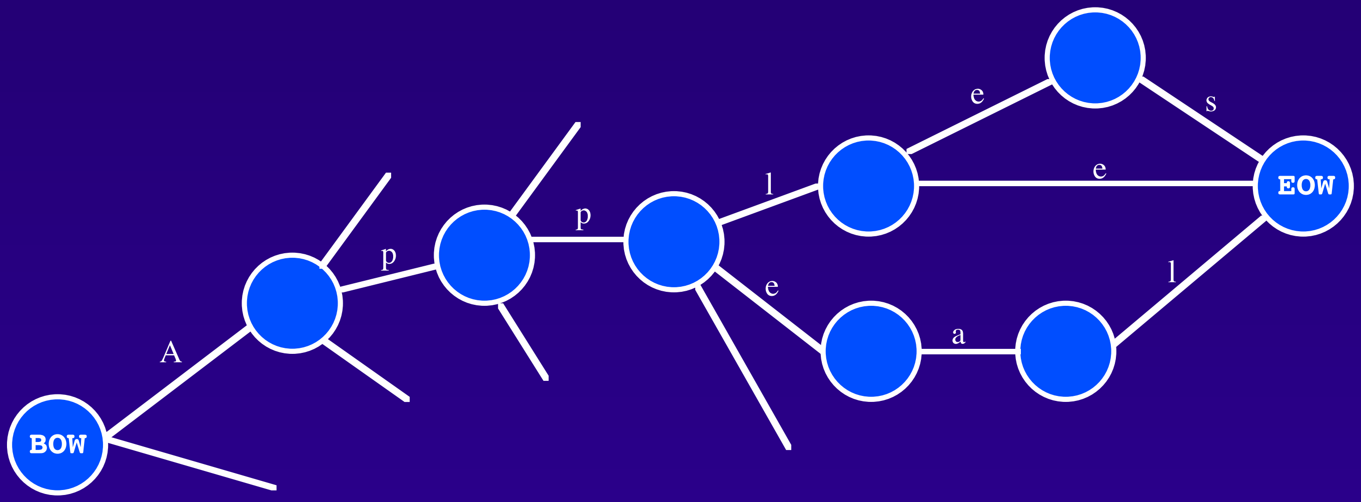
Lexical Trees
(The Wrong Way)

- Words Stored Separately



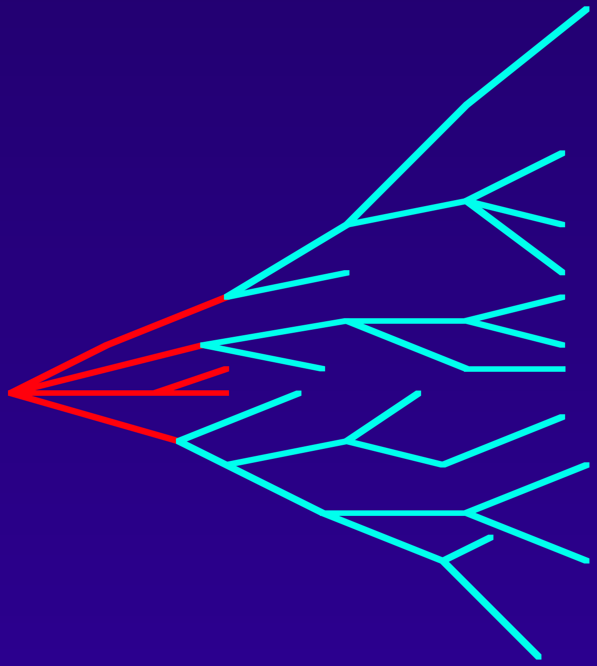
Lexical Trees
(The Right Way)

- Word Starts Merged Together



The Problem with Trees

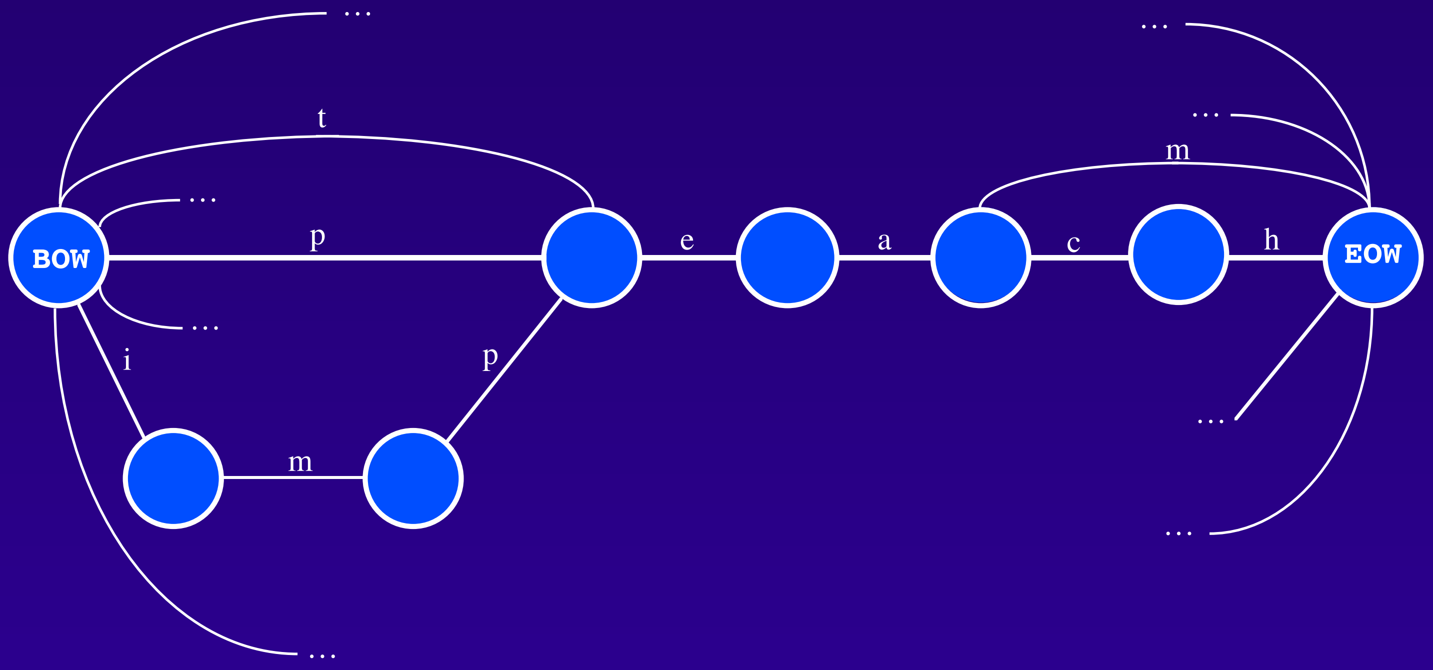
- Trees Are Compact at the Base...
- ... but Have Many Leaves



Lexical Graphs (Another Way)

- Word Endings Also Merged Together

  (e.g., team, teach, peach, impeach)



Consequences of Graph Convergence

- Probabilities Merged (or Discarded)
  - Currently Averaged if Retained
  - Threshold for Merging
  - Dicts Don't Care
- Exit Viterbi or N-Best
  - "met", "net", or "wet" May Be Three Top Choices
  - All But One Eliminated by Convergence to "...et"
  - Carry N Best Paths, Regardless of Node-Sharing
    - Beam Still Works

Creating Lexical Graphs

- Word Lists
  - With or Without Word-Frequencies
  - Newton Uses Dicts Exclusively (No Transition Probabilities)
  - Three-Tiered Word Classification
    - ~1000 Most Frequent Words
    - Few Thousand Moderately Frequent Words
    - Equivalent to ~100,000 Word Dictionary
      - Combined with Prefix & Suffix Dictionaries (For Alternate, Inflectional Forms)
  - Full Word- & Letter-Frequency Information Can Be Retained if Desired (But Are Not for Newton)

Creating Lexical Graphs

- Regular Expressions
    - Telephone Numbers Example:

```
dig    = [0123456789]
digm01 =   [23456789]

acodenums = (digm01 [01] dig)

acode  = { ("1-"?    acodenums "-"):40 ,
           ("1"? "(" acodenums ")"):60 }

phone = (acode? digm01 dig dig "-" dig dig dig dig)
```

Combining Lexical Graphs: "BiGrammars"

- Define Contexts as Probabilistic Combinations of Lexical Graphs

- Simple Telephone Context Example:

```
BiGrammar2 Phone

[phone.lang 1. 1. 1.]
```

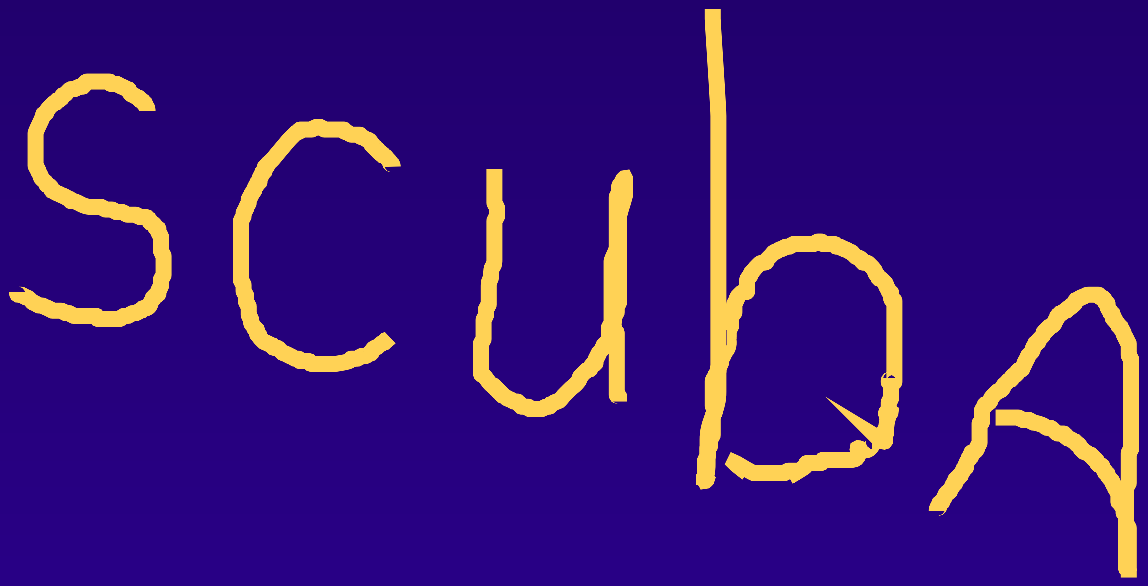
More Complex BiGrammar

```
BiGrammar2 FairlyGeneral
(.8
   (.6
      [WordList.dict .5   .8   1. EndPunct.lang .2]
      [User.dict     .5   .8   1. EndPunct.lang .2]
   )
   (.4
      [Phone.lang    .5   .8   1. EndPunct.lang .2]
      [Date.lang     .5   .8   1. EndPunct.lang .2]
   )
)

(.2
   [OpenPunct.lang  1.   0.   .5
      (.6
         WordList.dict .5
         User.dict     .5
      )
      (.4
         Phone.lang    .5
         Date.lang     .5
      )
   ]
)

[EndPunct.lang  0.  .9  .5  EndPunct.lang .1]
```

Geometric Context

- Estimates of Baseline, Topline, etc. Have Too Many Pathological Failure Modes
  - Produces Erratic Recognition Failures
- Use Relative Geometric Positions and Scaling Between Character Pairs ("GeoContext")

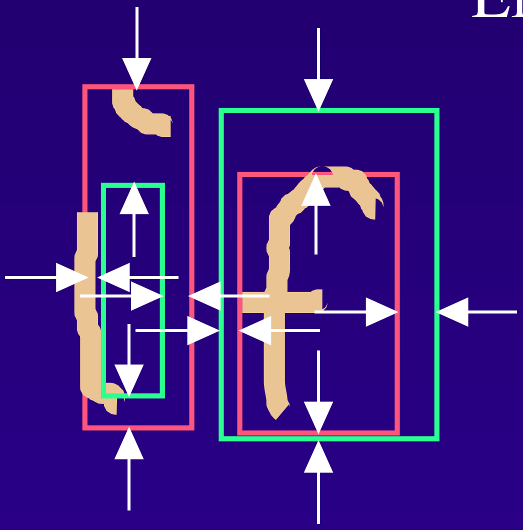
Recognition Ambiguity



GeoContext Example

“if” from User vs Table

Error Vector of
Eight Differences



(User Data Scaled to
Minimize Error Magnitude)

GeoContext Scoring

- Character Hypotheses Yield Expected Positions from Table
  - To Within a Scale Factor and Offset
    - User Data Scaled to Minimize Computed Error
  - Table is Learned in Data-Driven Process
- Error Vector is Computed
  - Modeled by Full Multi-Variate Gaussian Distribution for All Characters
- Quadratic Error Term Used as Score
  - Based on Inverse Grand Covariance Matrix

Old Newton Writing Example

when Year-old Arabian retire tipped off the Christmas wrap
No square with delights Santa brought the Attacking hit too dat
would Problem was, Joe talked Bobbie. His doll stones at the r
in its army Antiques I machine gun and hand decades At its side
it says things like 3 "Want togo shopping" The Pro has claimed
responsibility that's Bobbie Liberation Organization. Make up
more than 50 Concerned parents 3 Machinist 5 and oth er activi
the Pro claims to hsve crop if Housed switched the voice boxes
300 hit, Joe and Bobbie foils across the United States this holida
Season we have operations All over the country" said one pro
member 5 who wished to remain autonomous. "Our goal is to c
and correct Thu problem of exposed stereo in editorials toys."

ANHR Writing Example

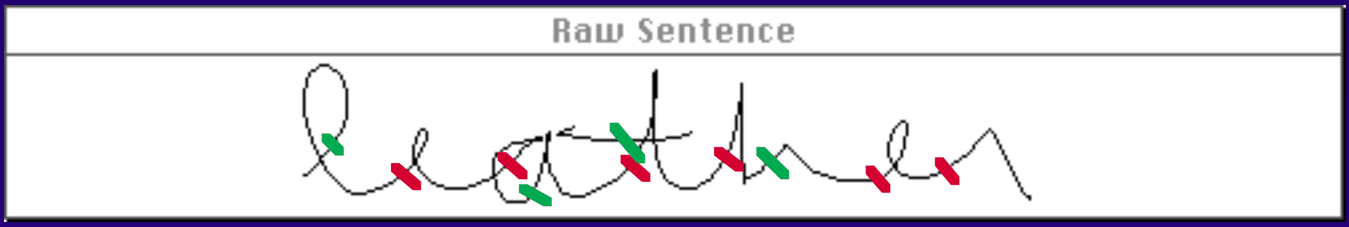
When 7-year-old Zachariah Zelin ripped off the Christmas wrapp
he squealed with delight. Santa brought the talking G.I.Joe doll
wanted. Problem was, Joe talked like Barbie. His doll stands at
ready in i ts Army fatigues, machine gun and hand grenades at i t
But it says things like, ll Want to go shopping?" The BLO has cl
responsibility. That's Parbie Liberation Organization. Made up
more than 50 concerned parents, feminists and other activists, the
claims to have surreptitiously switched the voice boxes on 3oo G
and Barbie dolls across the United States this holiday season. "W
have operatives all over the country," said one BLO member, wh
wished to remain anonymous. "Our goal is to reveal and correct
problem of gender-based stereotyping in children's toys!'

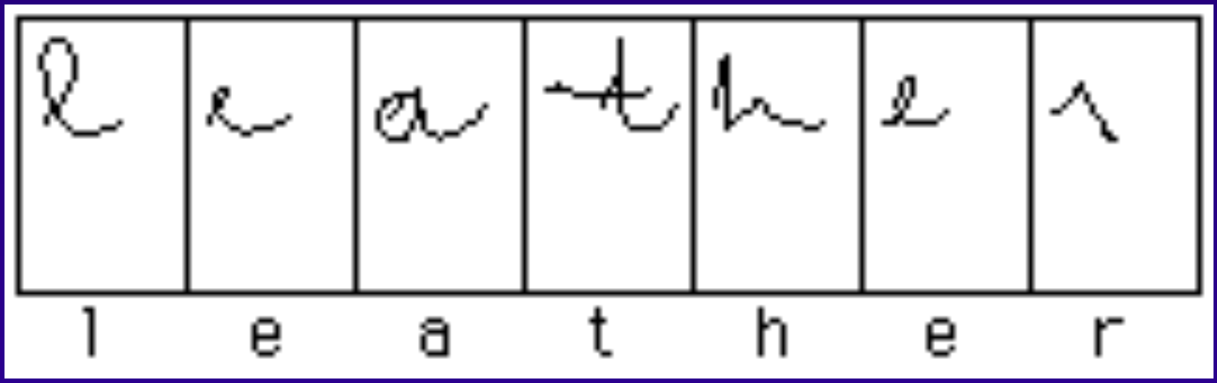
ATG Handwriting Recognition

ANHR Extensions

Cursive Handwriting

- Use Integrated Segmentation and Recognition with Stroke Fragments





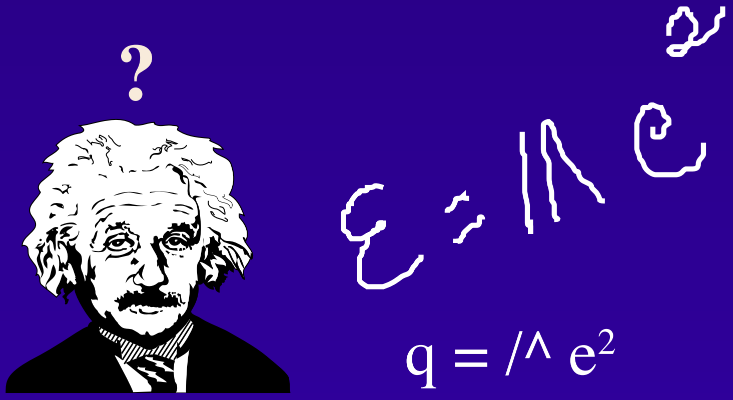
Chinese/Japanese/Korean

- Decompose Ideographic Characters ("Words") Into Radicals ("Characters") and Strokes, with Order and Placement Statistics
- Net Classifies “Alphabet” of About 300 Radicals
- Structure Lexicon in Terms of Legal Radical Sequences

User Independence vs. Adaptation

- Walk-Up Performance Drives In-Store Perception

- Individual Accuracy Drives Personal Use and Word of Mouth



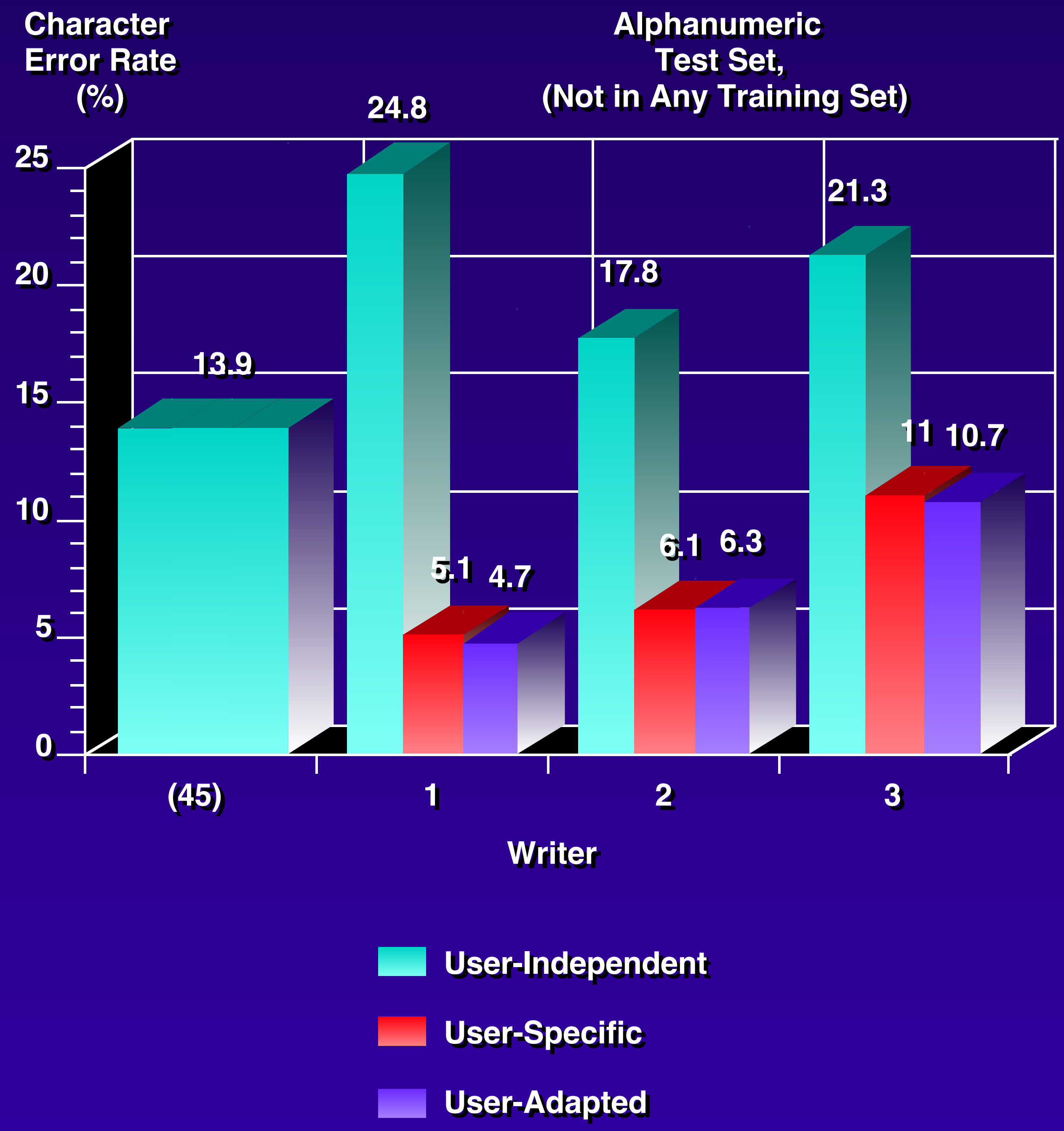
User Adaptation

- Neural Net Classifer Based On an Inherently Learning Technology
- Learning Not Used in Current Product Due to Memory Constraints
- User Independent “Walkup” Performance is Maintained!

User Adaptation

- User Training Scenario
  - 15-20 min. of Data Entry
    - Less for Problem Characters Alone
  - As Little as 10-15 minutes Network Learning
    - One-Shot Learning May Suffice
    - May Learn During Data Entry
    - Maximum of 2.5 hours (~12 Epochs)
- Learn on the Fly
  - Need System Hooks
  - Can Continuously Adapt!
  - Choosing What to Train On is Key System Issue

The Significance of Adaptation



The Power +o he your 6est