Finding Structure in Time

By Jonathan Hall

Author: Jeffrey L. Elman

Contents

- Problem
- Algorithm
- Results
- Conclusion

Problem

- Speech research
- Usual MLP takes all input at once and returns output all at once
- Speech needs temporal ordering

Past solutions to problem

- Usual approach: sequential representation of events
  - [0 1 1 1 0 0 0 0 0]
  - [0 0 0 1 1 1 0 0 0]
- Biological Implications
  - How to express parallel computations in terms of human brain?
- Not all input vectors same length
  - Example: translating user speech into text
- Applications: prediction problems

Solution to Problem

- Variable length input
- Analyze batches of events
- Dynamic system
- Remember past events

Algorithm

- Solution: give network memory with a recurrent network

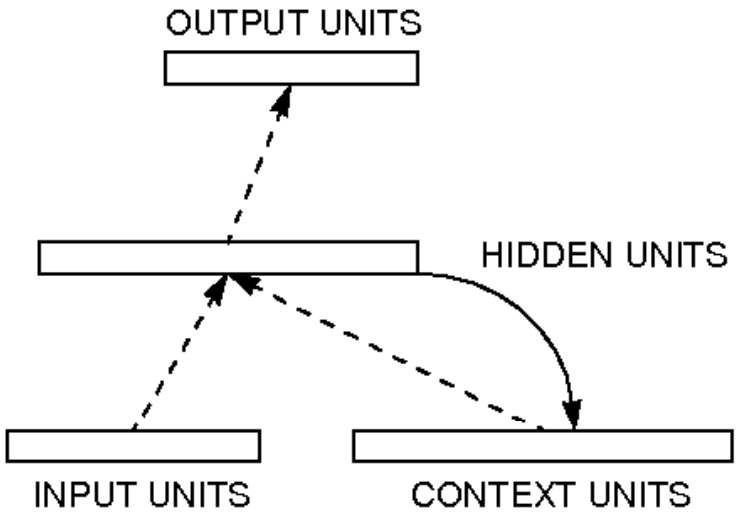
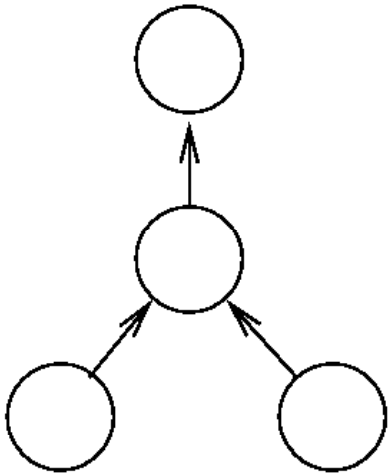
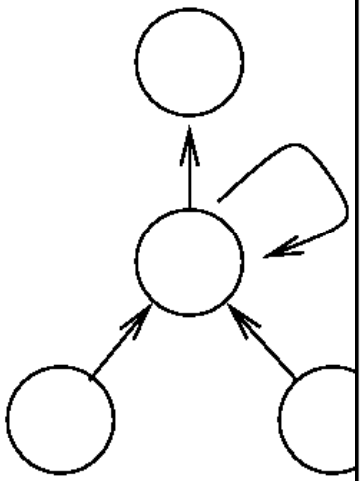


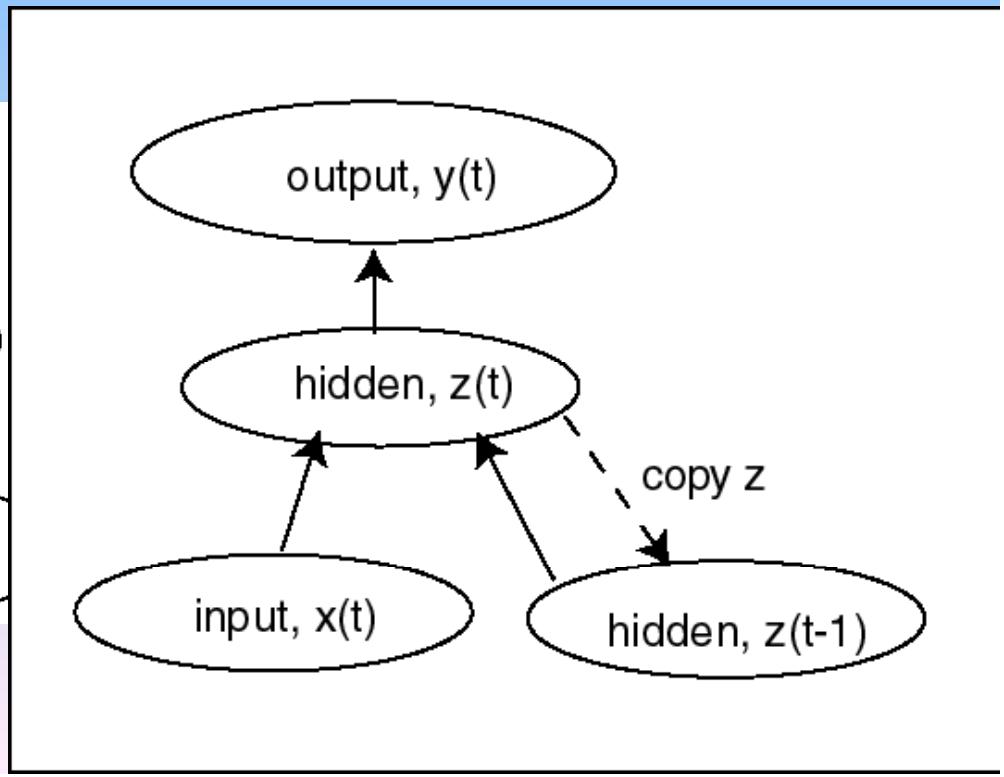
Figure 2. A simple recurrent network in which activations are copied from hidden layer to context layer on a one-for-one basis, with fixed weight of 1.0. Dotted lines represent trainable connections.

Recurrent Networks Overview

- **Has a directed cycle**
- Hopfield is a symmetric recurrent network
- Good for: learning grammar, speech recognition, and music composition







Author's Network

- Why did author use network on left and not the one on the right?

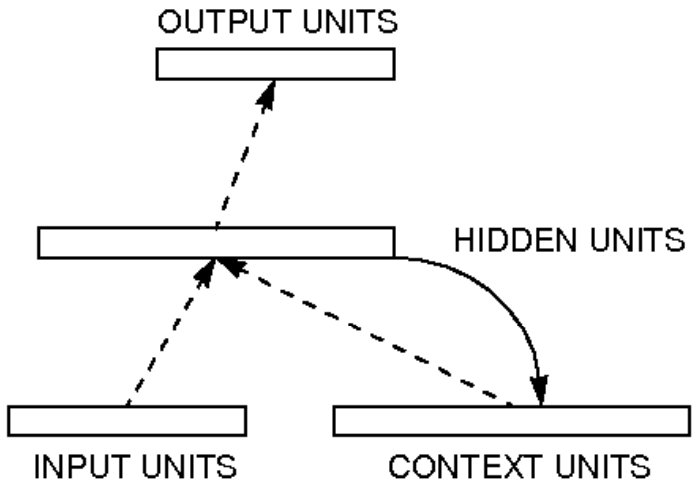
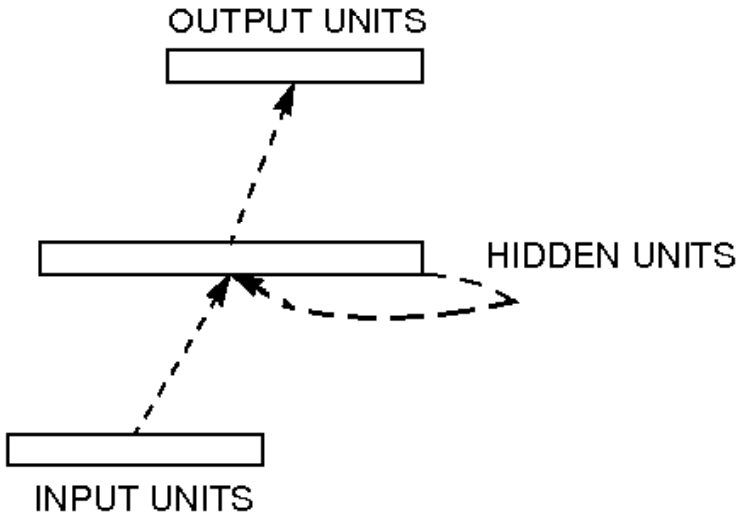


Figure 2. A simple recurrent network in which activations are copied from hidden layer to context layer on a one-for-one basis, with fixed weight of 1.0. Dotted lines represent trainable connections.



Results Contents

- XOR
- Letter Sequence
- Words
- Sentences

XOR

- Problem setup:
  - Input: 2 random bits, 3rd bit is XOR of previous 2 bits
  - Goal: predict next bit
  - Example Input: 1 0 **1** 0 0 **0** 0 1 **1** 1 1 **0** 1 0 **1**…

XOR Results

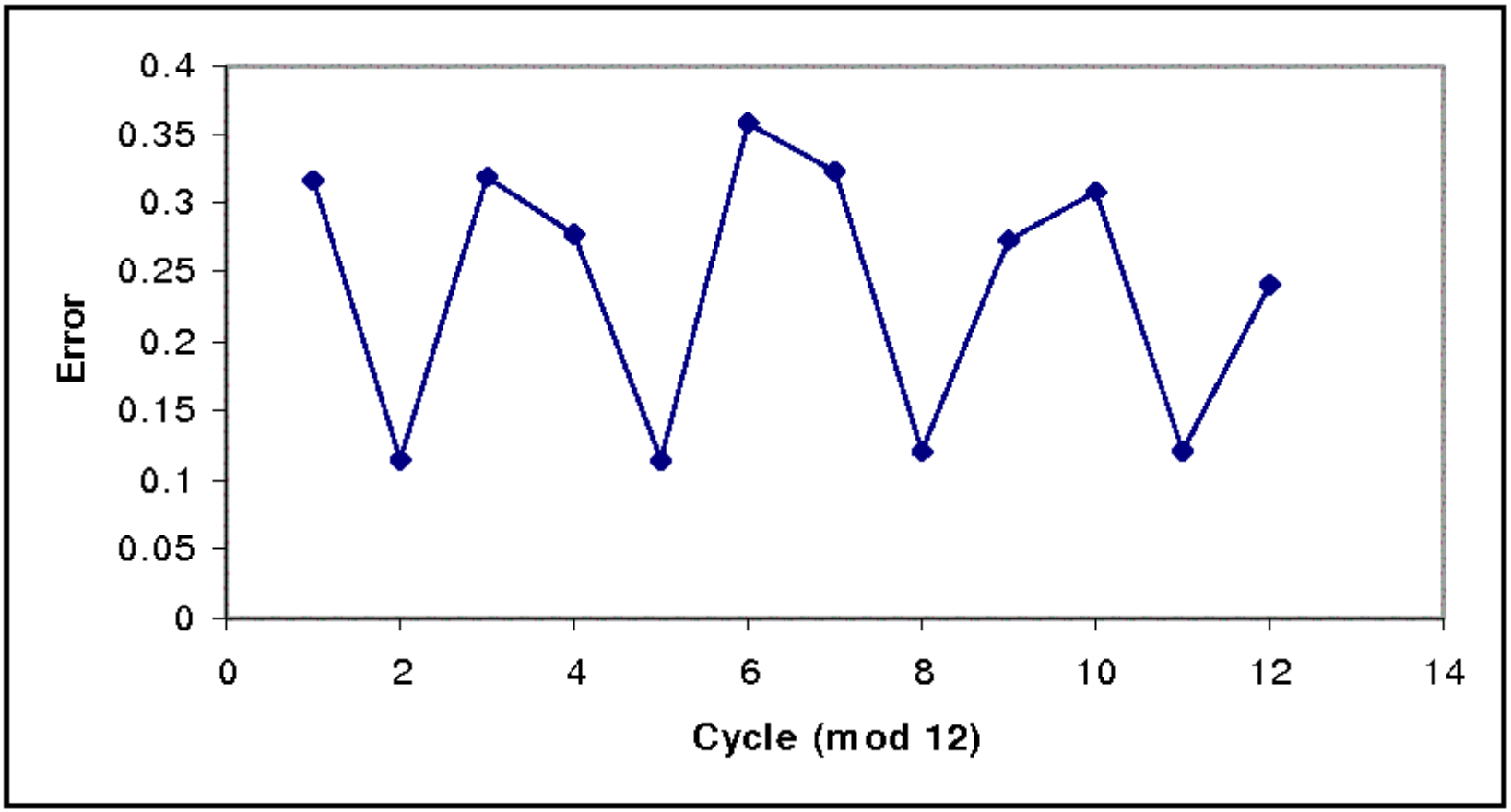


Figure 3. Graph of root mean squared error over 12 consecutive inputs in sequential XOR task. Data points are averaged over 1200 trials.

Letter Sequence

- Problem setup:
  - Input: Random consonants (b,d,g), then random consonants replaced with: b->ba, d->dii, g->guuu
  - Each letter given unique 1x6 vector
  - Goal: predict next letter

Table 4: Vector Definitions of Alphabet

	Consonant	Vowel	Interrupted	High	Back	Voiced
b	[1	0	1	0	0	1]
d	[1	0	1	1	0	1]
g	[1	0	1	0	1	1]
a	[0	1	0	0	1	1]
i	[0	1	0	1	0	1]
u	[0	1	0	1	1	1]

Letter Sequence Results

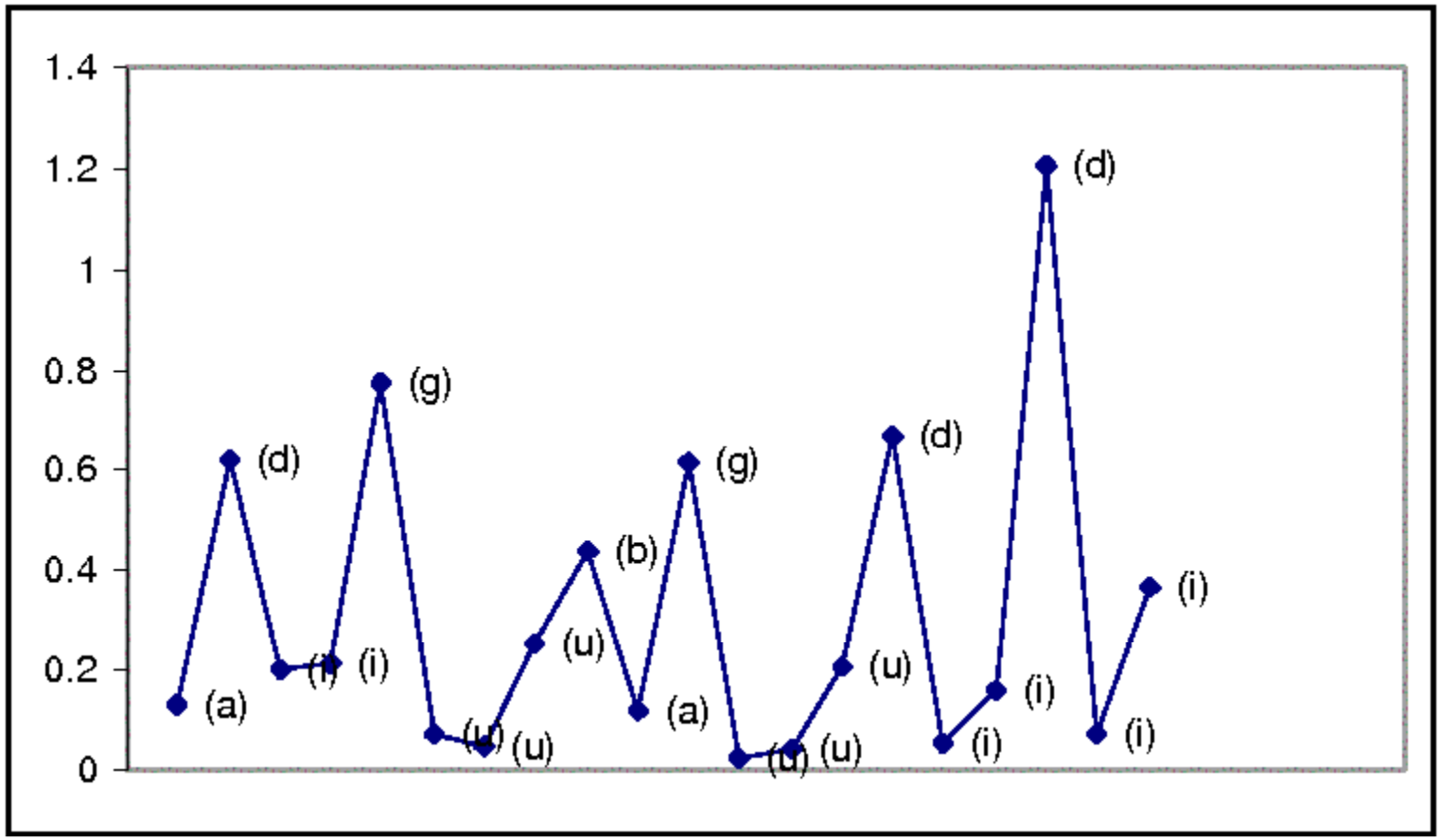


Figure 4. Graph of root mean squared error in letter prediction task. labels indicate the correct output prediction at each point in time. Error is computed over the entire output vector.

Words

- Problem Setup:
  - Input: String of sentences with no breaks between words or sentences
  - Goal: predict next letter of sequence

Words Results

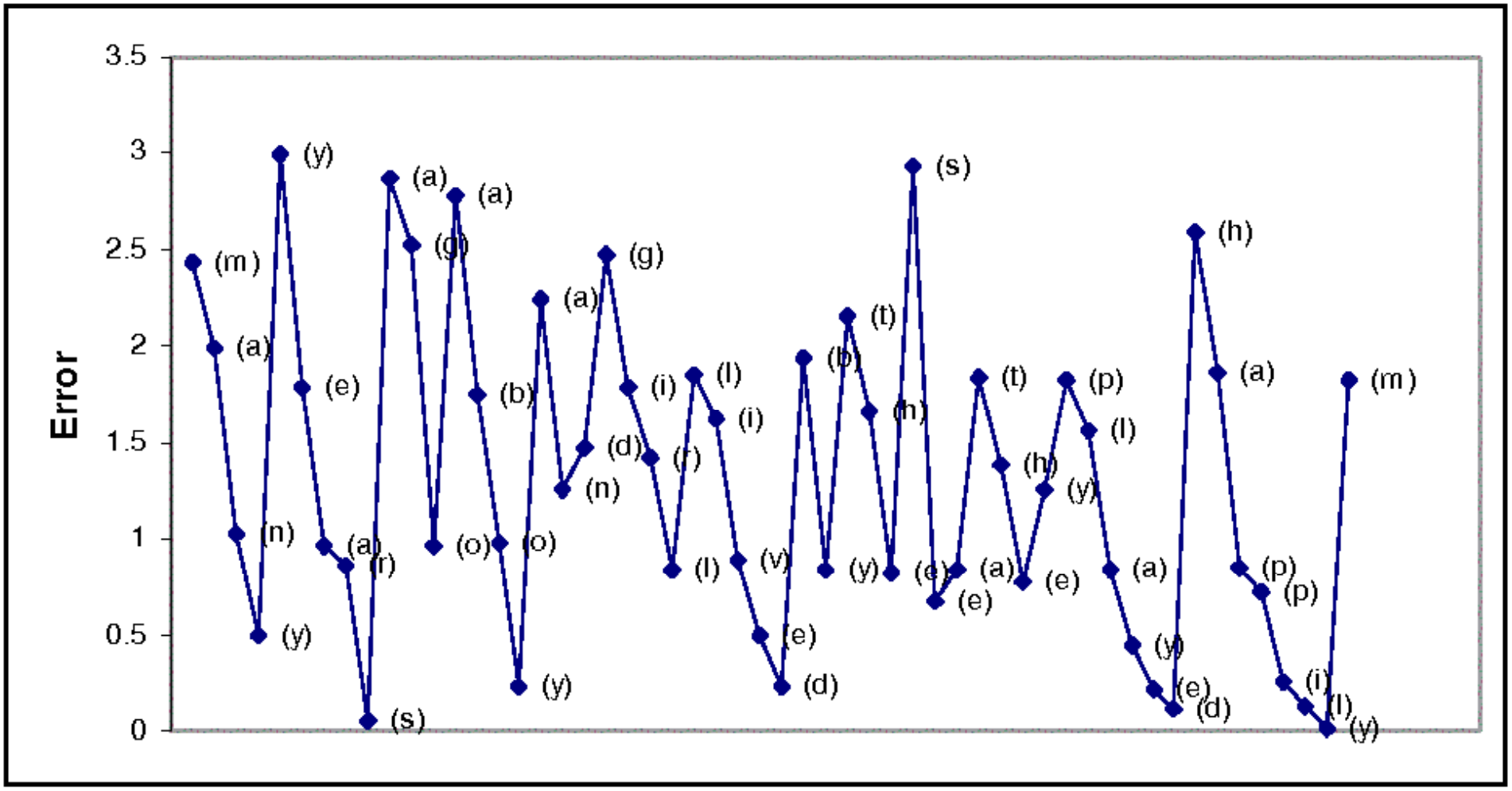


Figure 6. (a) Graph of root mean squared error in letter-in-word prediction task.

Sentence Setup

- A large number of simple sentences were randomly produced
- Each word was vectorized
- No breaks in between sentences
- Goal: predict the next word

Sentences

Table 4: Templates for sentence generator

WORD 1	WORDS	WORD 3
NOUN-HUM	VERB-EAT	NOUN-FOOD
NOUN-HUM	VERB-PERCEPT	NOUN-INANIM
NOUN-HUM	VERB-DESTROY	NOUN-FRAG
NOUN-HUM	VERB-INTRAN	
NOUN-HUM	VERB-TRAN	NOUN-HUM
NOUN-HUM	VERB-AGPAT	NOUN-INANIM
NOUN-HUM	VERB-AGPAT	
NOUN-ANIM	VERB-EAT	NOUN-FOOD
NOUN-ANIM	VERB-TRAN	NOUN-ANIM
NOUN-ANIM	VERB-AGPAT	NOUN-INANIM
NOUN-ANIM	VERB-AGPAT	
NOUN-INANIM	VERB-AGPAT	
NOUN-AGRESS	VERB-DESTORY	NOUN-FRAG
NOUN-AGRESS	VERB-EAT	NOUN-HUM
NOUN-AGRESS	VERB-EAT	NOUN-ANIM
NOUN-AGRESS	VERB-EAT	NOUN-FOOD

Table 3: Categories of lexical items used in sentence simulation

Category	Examples
NOUN-HUM	man, woman
NOUN-ANIM	cat, mouse
NOUN-INANIM	book, rock
NOUN-AGRESS	dragon, monster
NOUN-FRAG	glass, plate
NOUN-FOOD	cookie, sandwich
VERB-INTRAN	think, sleep
VERB-TRAN	see, chase
VERB-AGPA	move, break
VERB-PERCEPT	smell, see
VERB-DESTROY	break, smash
VERB-EA	eat

Sentences: quality measurement

- Can’t use word-by-word RSS
  - Error: 0.88
- Solution: use RSS for categories of words
  - Error: 0.053
- How did the network do this?

Sentence Classification

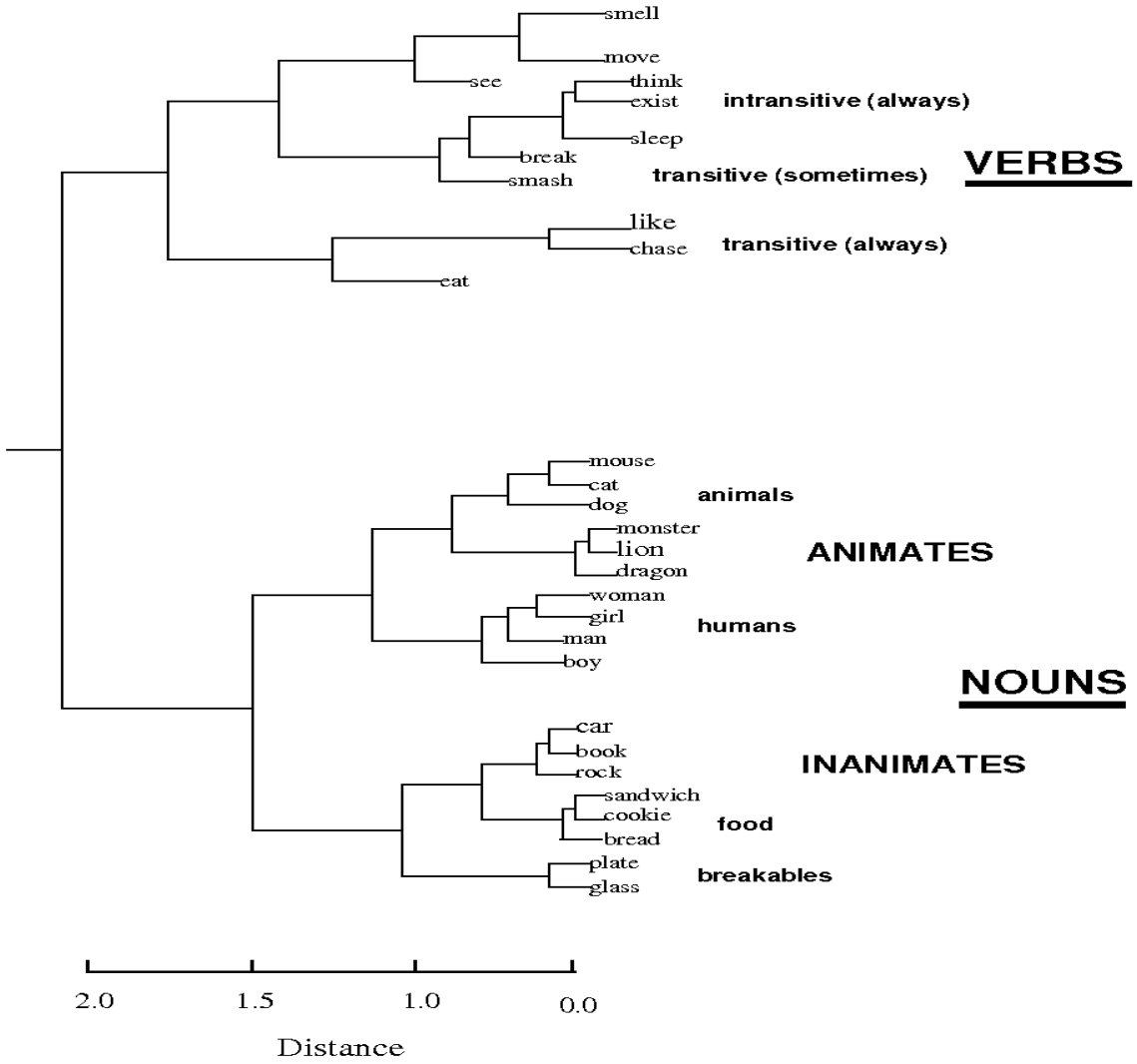


Figure 7. Hierarchical cluster diagram of hidden unit activation vectors in simple sentence prediction task. labels indicate the inputs which produced the hidden unit vectors; inputs were presented in context, and the hidden unit vectors averaged across multiple contexts.

Conclusion

- Some problems are different when expressed as temporal events.
- RSS can be used to analyze the temporal structure.
- Length of sequential dependencies doesn’t always worsen performance.
- Representation of time and memory is task-dependant.
- Representations can be structured.

The End