Investigating systematicity in the linear RAAM neural network

Igor Farkaš & Michal Pokorný
Department of Applied Informatics
Comenius University in Bratislava
Slovak Republic

DFKI, Saarland University, Saarbruecken, November 2009

Presented at: *Neural Computation & Psychology Workshop*, Oxford, UK, July 2008
Published in: Mayor J. et al (Eds.), *Connectionist Models of Behaviour and Cognition*, Vol. II, Singapore: World Scientific, pp. 217-228, 2009

Talk outline

- Introduction to systematicity
- Processing structured data (symbolic, connectionist)
- Features of connectionist processing
- Linear RAAM
- testing systematicity with linear RAAM using various types of word representations
- conclusion

Systematicity argument

Systematicity – a number of putative psychological properties (regularities)

Language of thought (J. Fodor) - the view that all mental representations are linguistic expressions within an “internal” language which significantly resembles spoken language.

Human thought is systematic because

ability to think certain thoughts (*X eats Y*) implies the ability to think others (*Y eats X*), thoughts are intrinsically connected

Human thought / language is

productive (-> generativity)

compositional (concatenative c.)

Criticism of connectionist models (Fodor & Pylyshyn, 1988): NNs use mental representations that lack syntactic and semantic combinatorial structure, therefore NN cannot account for human cognition.

Systematicity in language

Systematicity arises automatically in symbolic computational models, but is not so straightforward in connectionist models (neural networks).

Systematic behaviour → various degrees of generalization

Two kinds:

Syntactic – judging grammaticality of novel sentence

Semantic – producing semantic representation of novel sentence

syntactic s. is easier to operationalize

semantic systematicity: what kind of semantic reps to use?

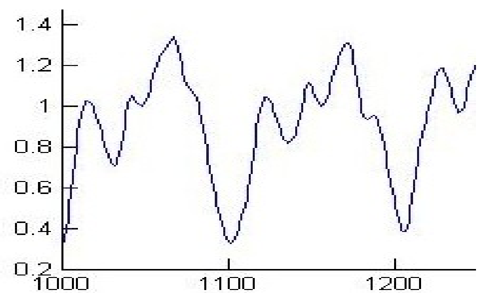
Data types

Numeric | Symbolic

vectorial

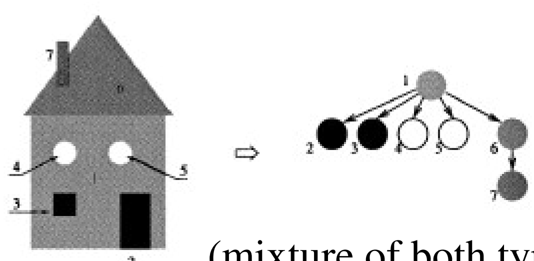
alpha, beta

sequences

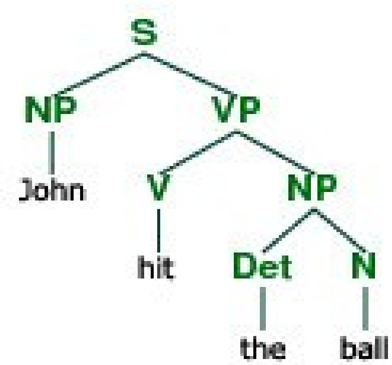


John hit the ball

n-ary trees



(mixture of both types)



Data processing methods

symbolic

high-level (decision trees, logic programming,...)

transparent, understandable, but difficult to adapt

subsymbolic (numeric)

low-level (clustering algorithms, neural nets,...)

more difficult to understand but

(relatively) well adaptable

Nature of cognitive (mental) representations remains a hot issue:

Propositions, with amodal symbols (Fodor & Pylyshyn, Landauer)	vs	Embodied, perception-based, modal (Barsalou, Lakoff, Glenberg)

Computational processing of structured data

symbolic

explicit manipulation of constituents (symbols)

no memory problem

no problem with recursion

no robustness

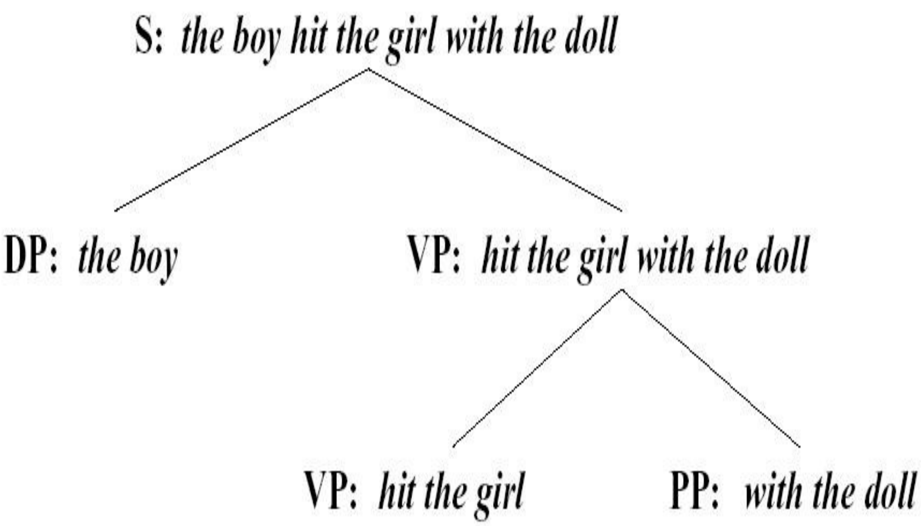
subsymbolic (connectionist)

no pointers

soft capacity limits for recursion and memory

robustness

higher cognitive plausibility



Challenge in cognitive science – NN approaches that are not *mere implementations* of symbolic models

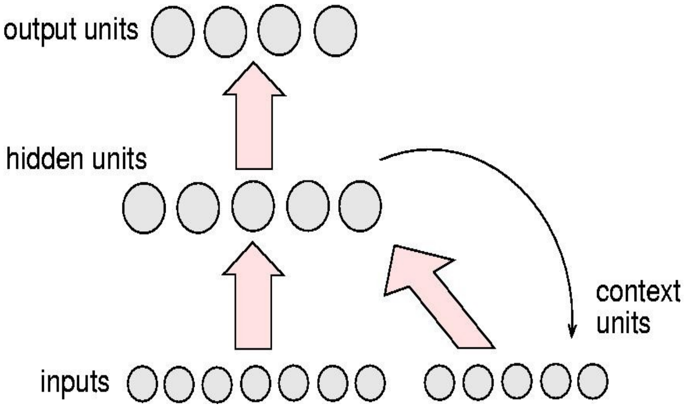
Processing structured data with neural nets

Sequence processing

recurrent neural networks (e.g. Elman net)

next-item prediction

error gradient-based learning (supervised)

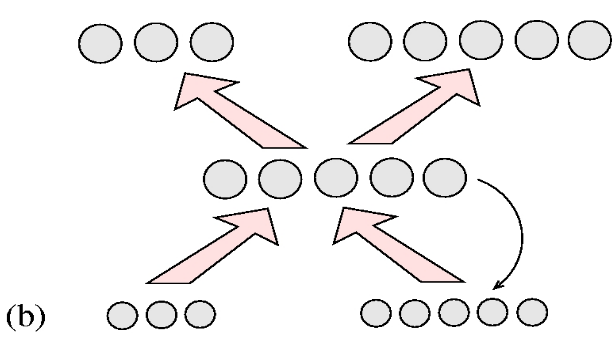


Tree processing

Recursive Auto-Associative Memory

tree encoding/decoding

self-supervised



There exist also impressive, complex NN models: e.g. INSOMNet (Mayberry, 2003; Rohde, 2002)

Handling compositionality with neural networks

- structured (composite) data around us

Hammer (2005):

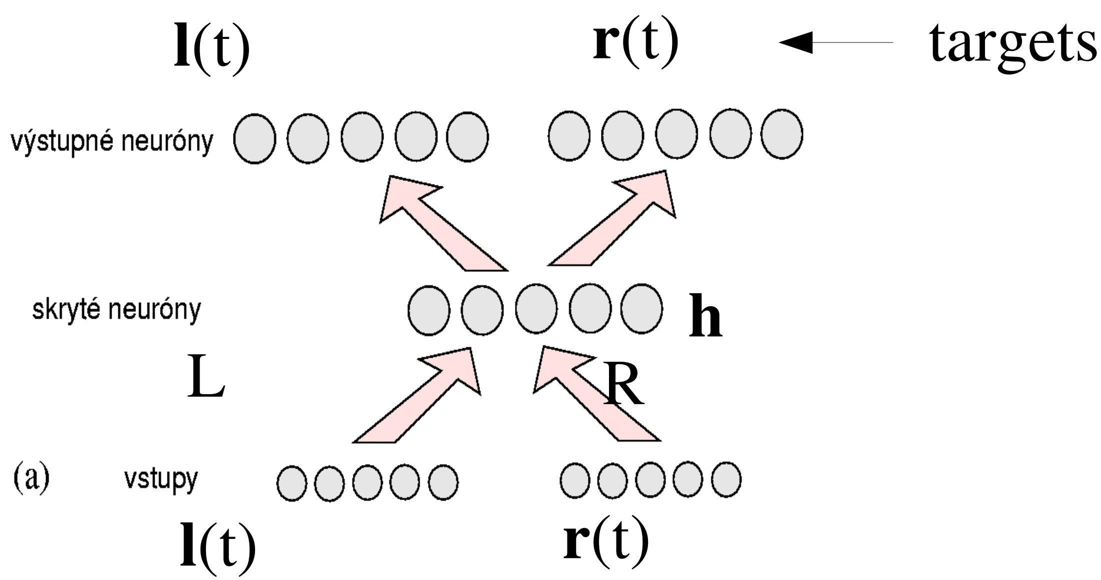
Which neural dynamics allow composite structures, grouping and binding to emerge in such a way that the single parts can be restored rapidly and, at the same time, the whole composite structure can be identified with a single object?

How can we adapt standard neural techniques to composite structures?

Recursive Auto-Associative Memory

Binary RAAM

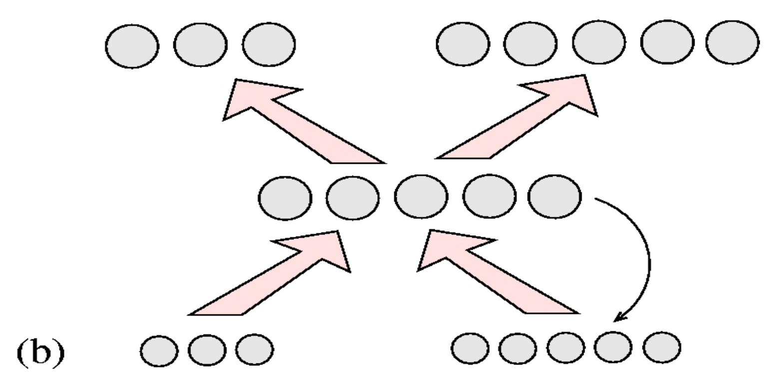
- suitable for trees



Compressed activation: $\mathbf{h}(t) = f\,(\mathbf{L}\mathbf{l}(t) + \mathbf{R}\mathbf{r}(t))$

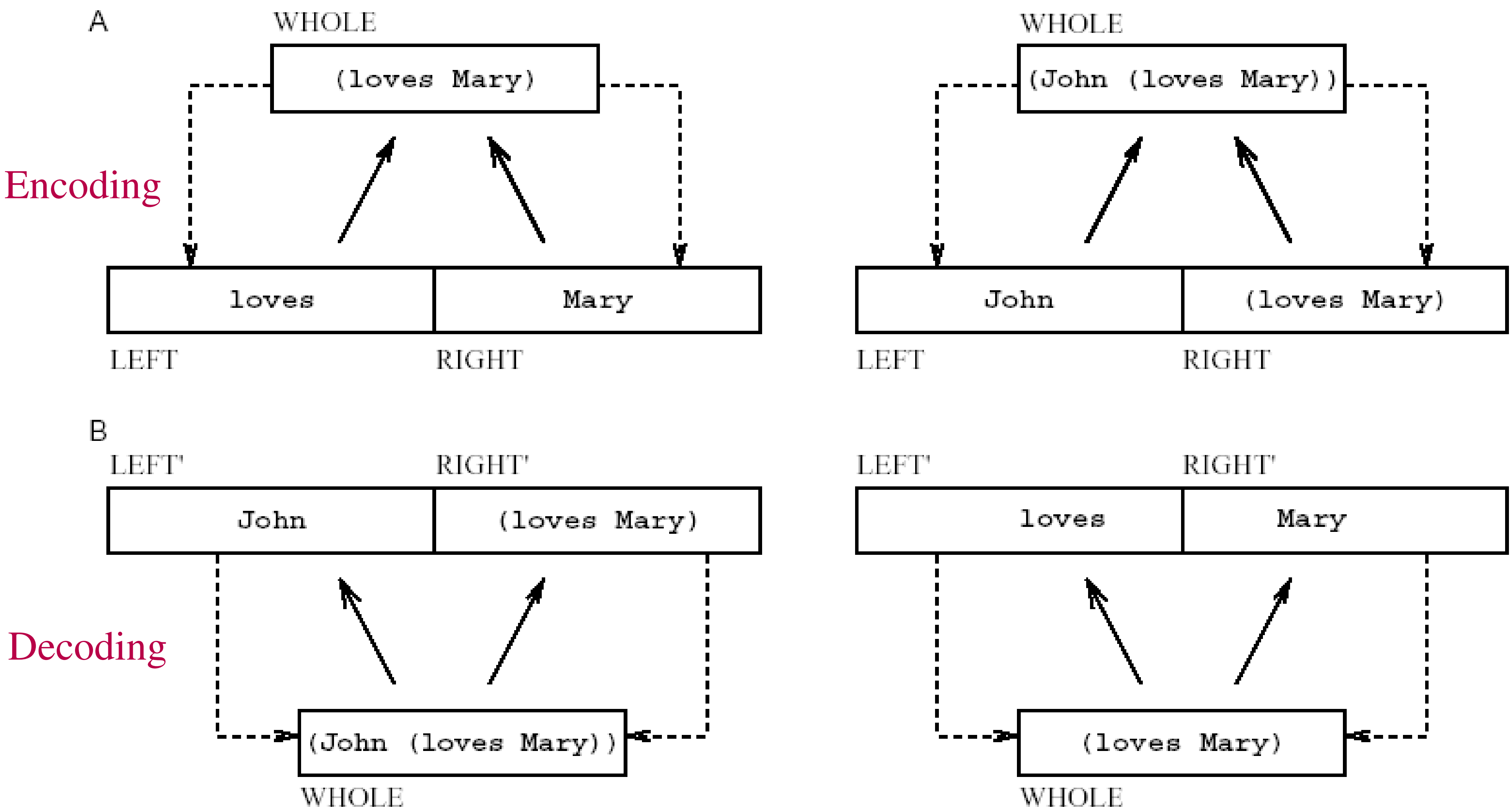
Sequential RAAM

- suitable for sequences



Tree encoding and decoding in RAAM

(John (loves Mary))



(Binary) Linear RAAM

Voegtlin (2002, 2005)

linear one-layer network - decoding uses transposed weight matrices

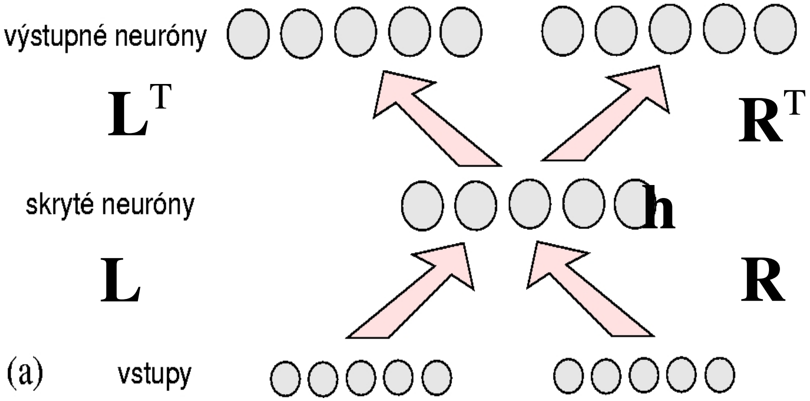
$$h_i(t) = \Sigma_j L_{ij}\, l_j(t) + \Sigma_j R_{ij}\, r_j(t)$$

uses unsupervised learning - resembles principal components analysis (PCA)

$$\Delta L_{ij} = \eta\, h_i(t)\, [l_j(t) - \Sigma_l L_{lj}\, w_l(t)]$$

$$\Delta R_{ij} = \eta\, h_i(t)\, [r_j(t) - \Sigma_l R_{lj}\, w_l(t)]$$

- moving target problem



Terminal test for decoding

parameter: chosen reconstruction threshold (theta)

Possible testing cases:

ambiguous test

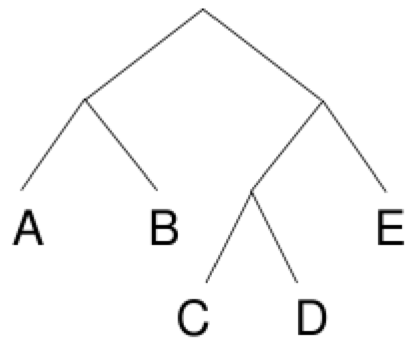
unrecognized terminal (infinite loops possible)

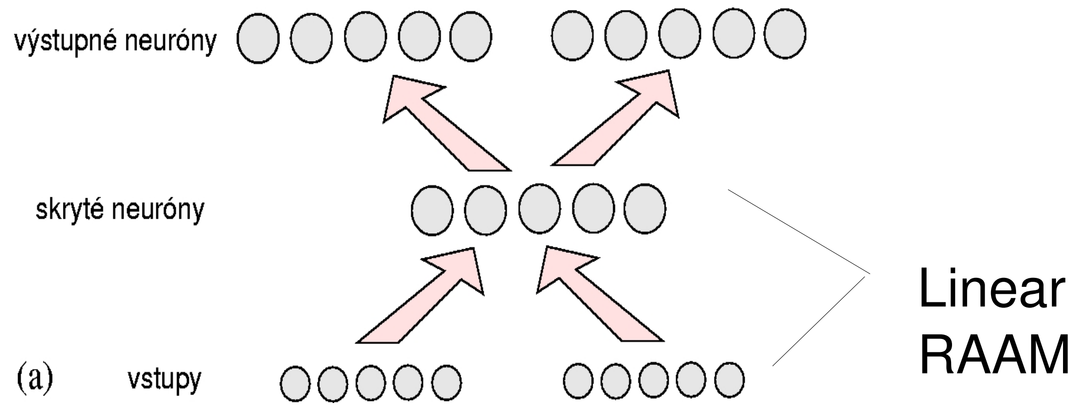
terminating non-terminal

wrong terminal

Decoding (of a tree) is considered successful if <u>none</u> of the above cases occurs. Then, a tree is representable by the network.

Example: Encoding a tree





Input $\mathbf{z}$		Reduced repr.		Output $\bar{\mathbf{z}}$
(A B)	$\longrightarrow$	$R_{AB}(t_1)$	$\longrightarrow$	$(\bar{A}\ \bar{B})$
(C D)	$\longrightarrow$	$R_{CD}(t_2)$	$\longrightarrow$	$(\bar{C}\ \bar{D})$
$(R_{CD}(t_2)\ E)$	$\longrightarrow$	$R_{CDE}(t_3)$	$\longrightarrow$	$(\bar{R}_{CD}(t_2)\ \bar{E})$
$(R_{AB}(t_1)\ R_{CDE}(t_3))$	$\longrightarrow$	$R_{ABCDE}(t_4)$	$\longrightarrow$	$(\bar{R}_{AB}(t_1)\ \bar{R}_{CDE}(t_3))$

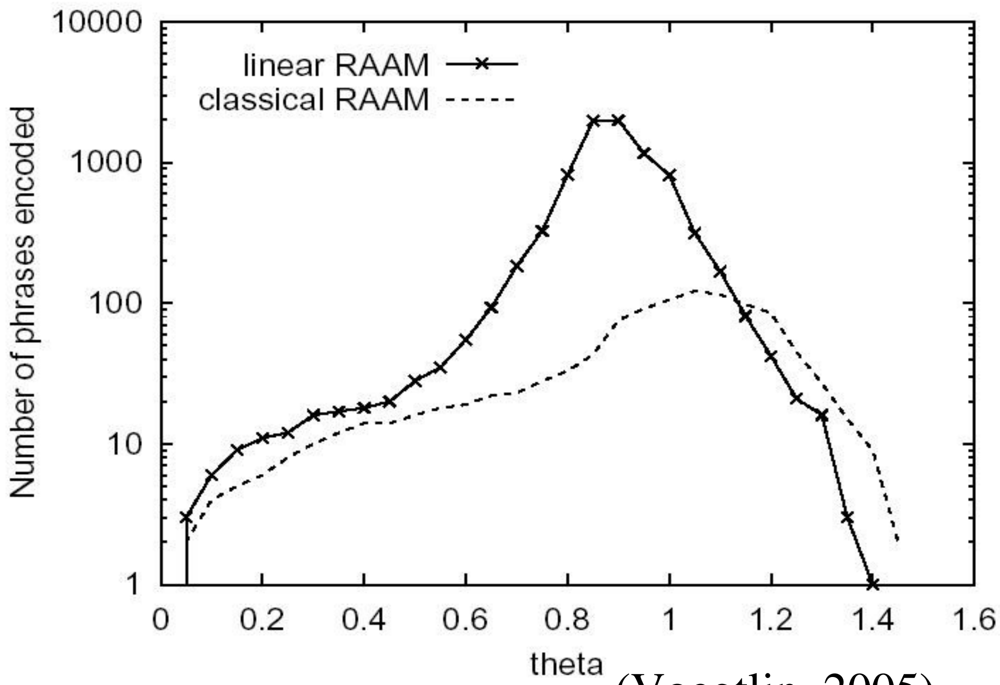
Linear RAAM: syntactic trees experiment

Sentence	*Noun Phrase*	*Verb Phrase*	*Prep. Phrase*	*Adj. Phrase*
S → NP VP	NP → d AP	VP → v NP	PP → p NP	AP → a AP
S → NP v	NP → d n	VP → v PP		AP → a n
	NP → NP PP			

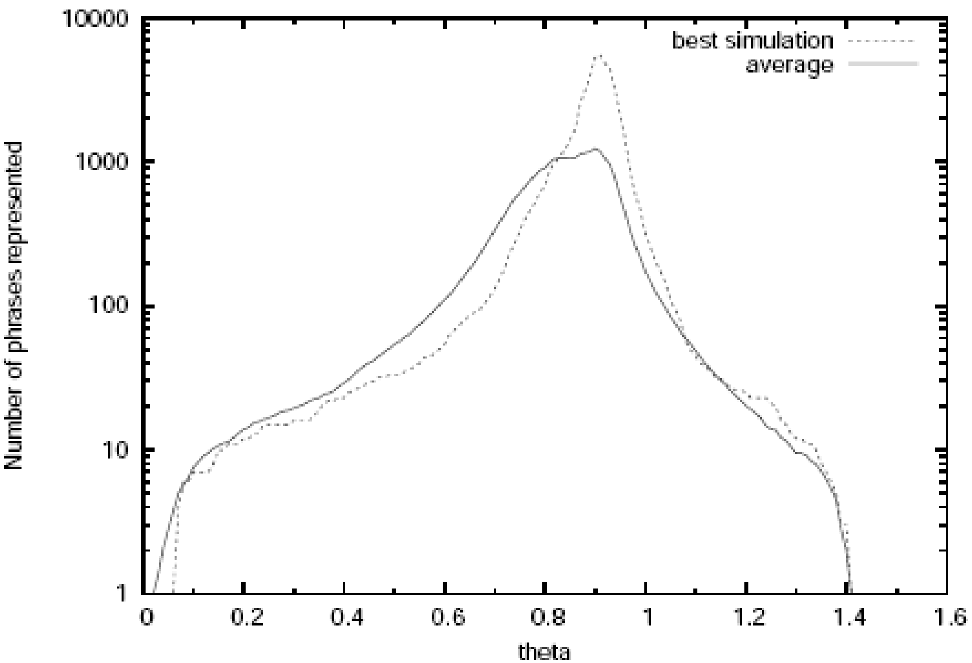
20-10-20 net

one-hot encoding of terminals

Generalization test replicated



(Voegtlin, 2005)



(Farkas & Pokorny, 2008)

Effect of terminal encoding on generalization

Lexicon *N* = 50 words, 100 sentences generated using a prob. CFG

neutral (one-hot) encoding (*N*-dim. Vectors) – 1 unit for 1 symbol

WCD-based encoding:

word co-occurrence detector (Li, Farkas, MacWhinney, 2004)

*N*x*N* contingency table (i-th rows and i-th column => 2*N* dim.)

rescaled (x2) components to enhance discrimination

WordNet-based word features:

Harm's (2002) feature generation system

1 to 6 binary features per word (hypernyms, features)

Training data - sentences with stochastic CFG

Sentences

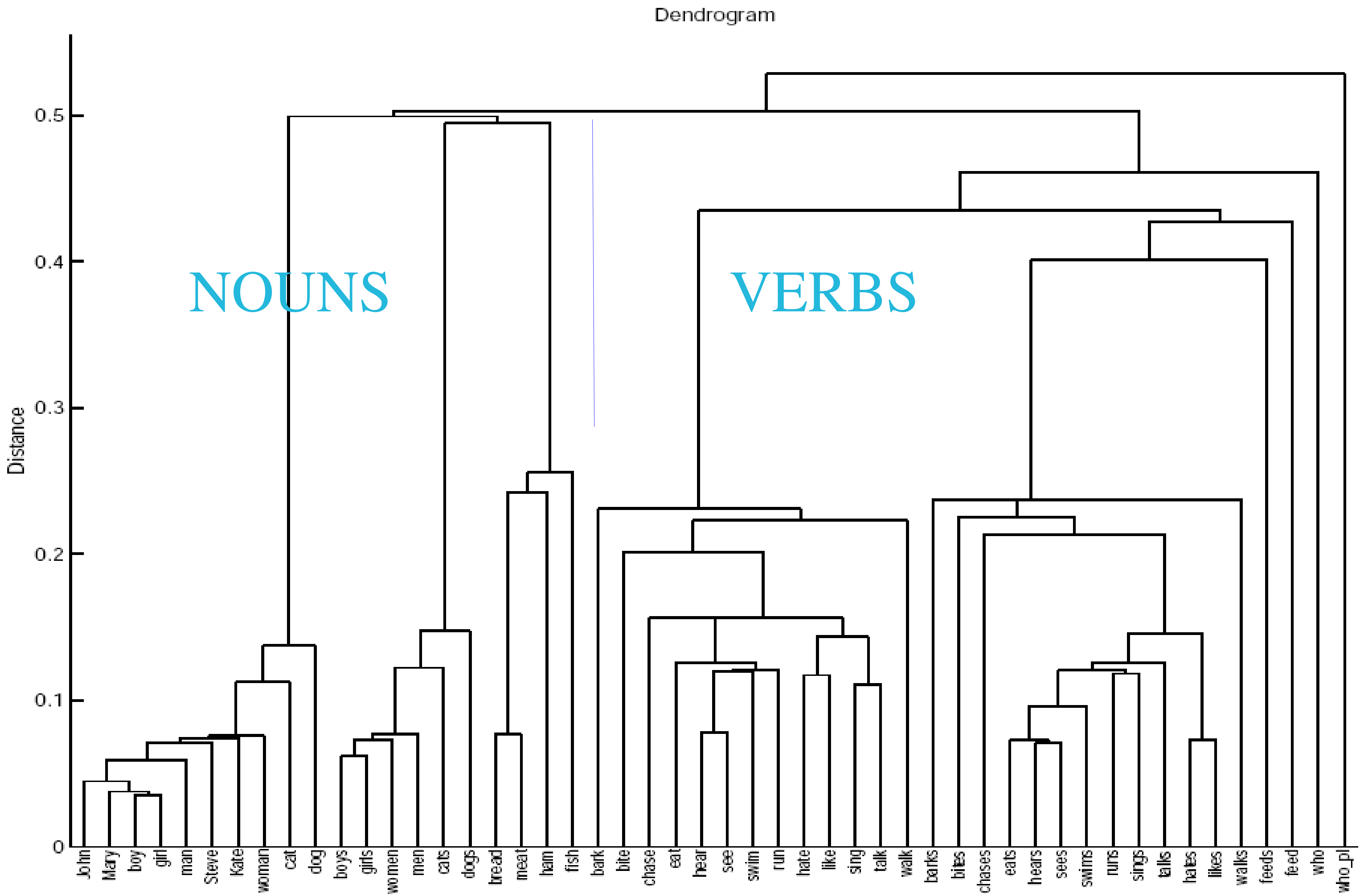
a) Steve walks
b) women see boys
c) dogs who_pl see girl bark
d) boy feeds cat who John sees
e) dog who boy sees sees cat who chases cats
f) boy who girl likes walks dogs who_pl see John who hates cat
g) cat who sees boys who_pl like girl who runs walks

Corresponding (ternary) trees

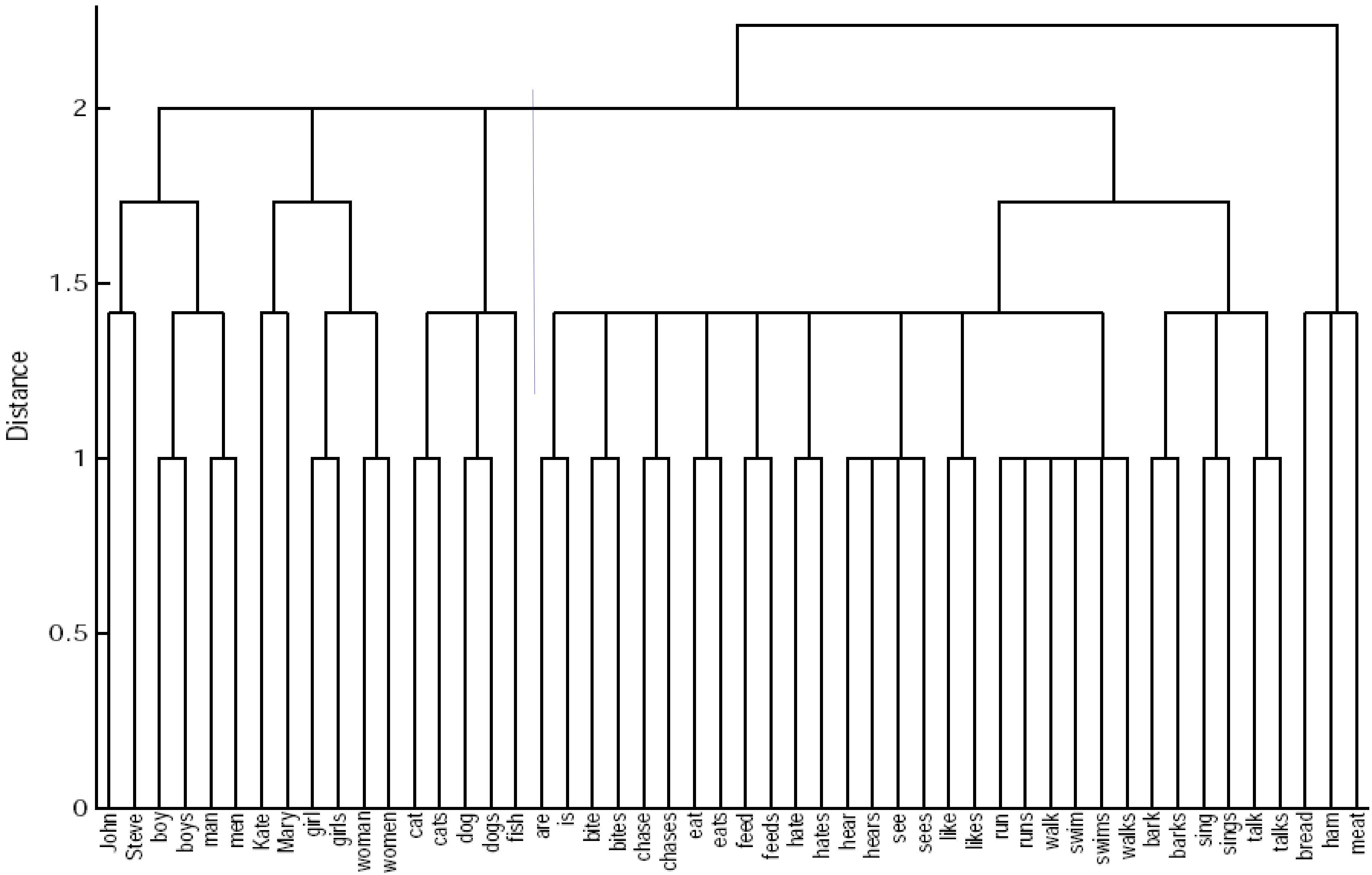
(ACTION AGENT OBJECT)

a) (walks Steve NULL)
b) (see women boys)
c) (bark (are dogs (see dogs girl)) NULL)
d) (feeds boy (is cat (sees John cat)))
e) (sees (is dog (sees boy dog)) (is cat (chases cat cats)))
f) (walks (is boy (likes girl boy)) (are dogs (see dogs (is John (hates John cat)))))
g) (walks (is cat (sees cat (are boys (like boys (is girl (runs girl NULL)))))) NULL)

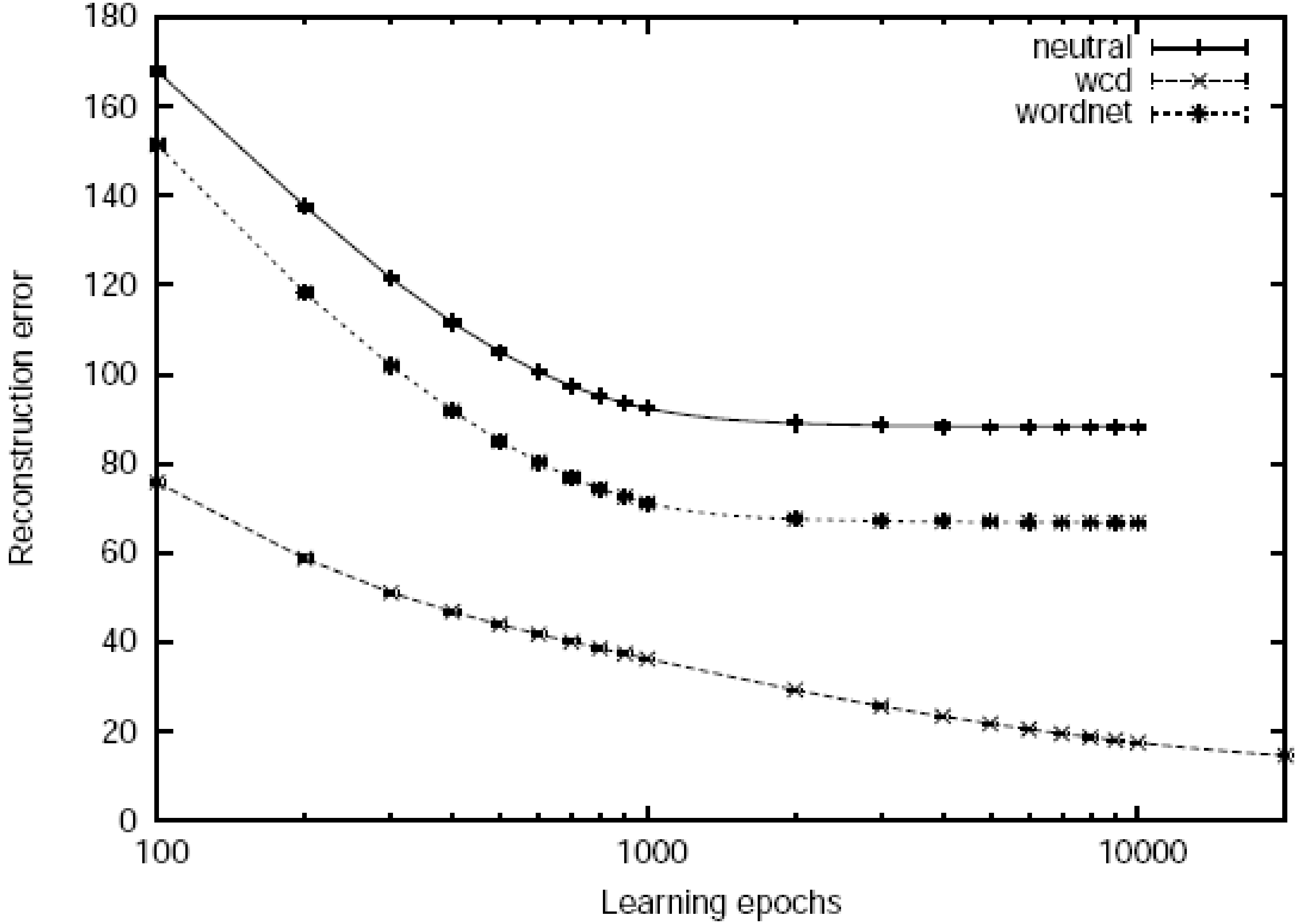
Dendrogram of WCD-based word codes



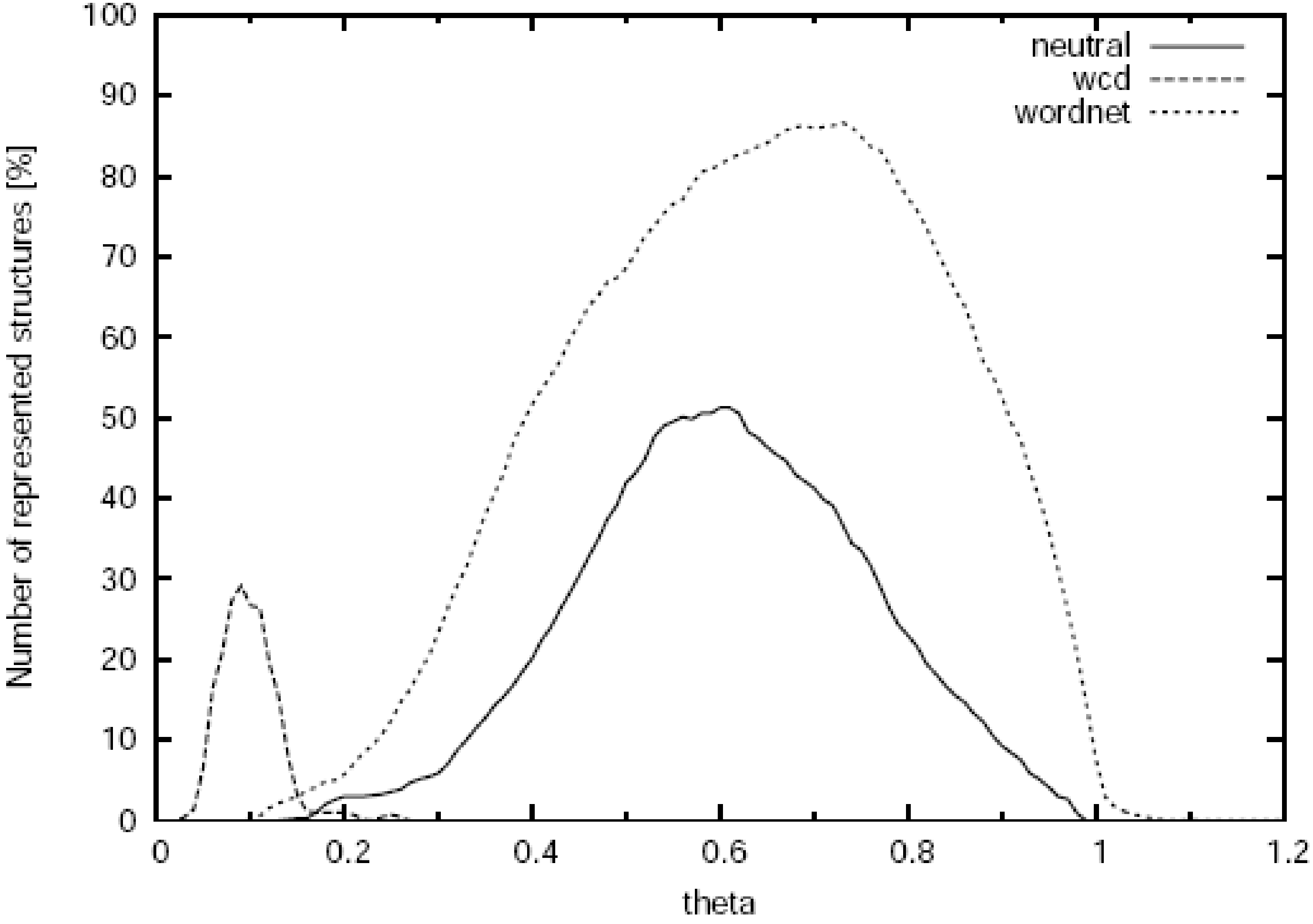
Dendrogram of WordNet-based word codes



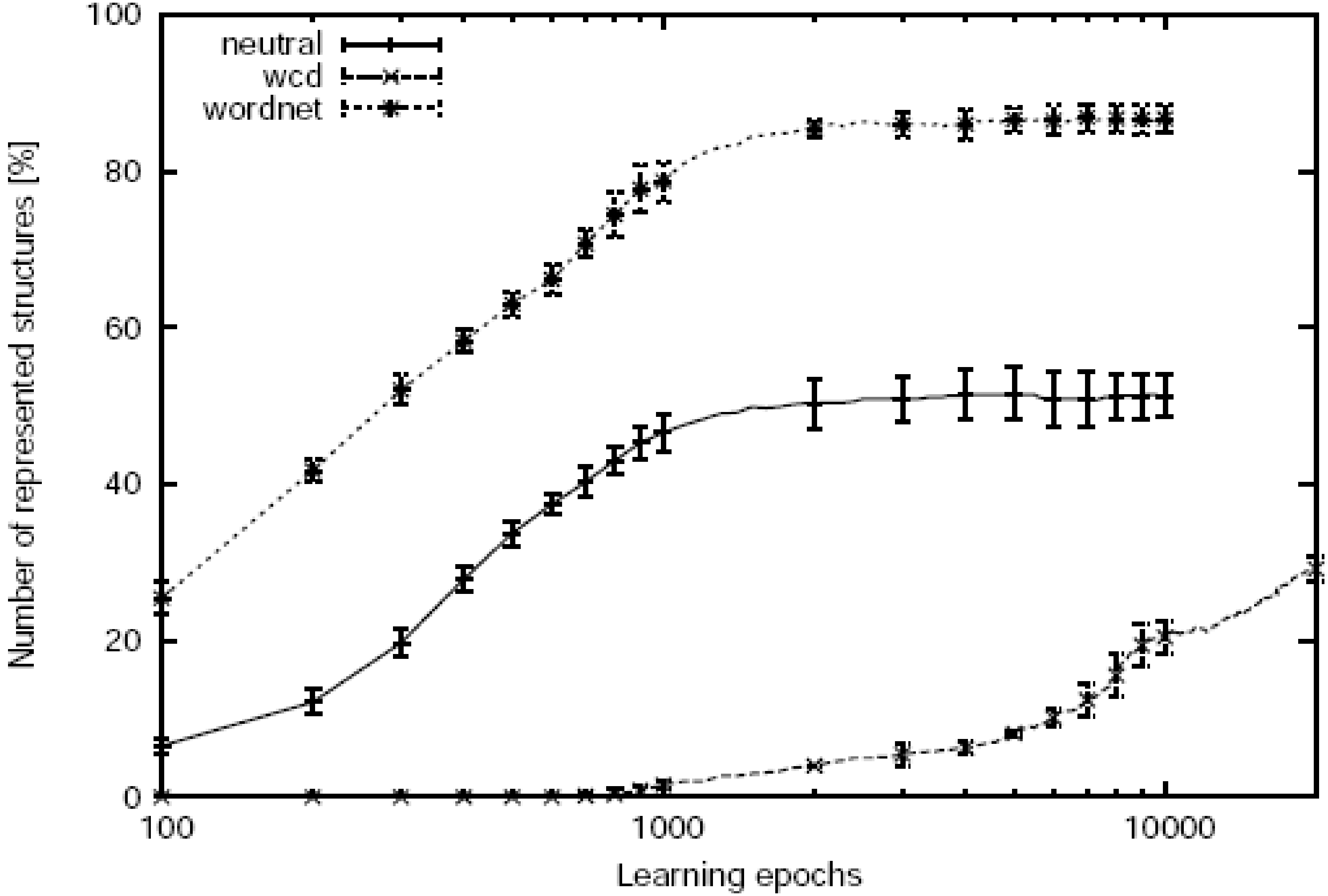
Effect of training on reconstruction error



Effect of reconstruction threshold on performance



Effect of training on performance using optimal theta



Testing levels of systematicity

Niklasson & van Gelder (1994)

0 – No novelty (training data used)

1 - Novel sentences (all atoms occurred in training in the known positions)

e.g. Trn: A sees B, B loves A, A sees B,   test: B sees A.

2 - Novel positions (at least one atom appears in new syntactic positions)

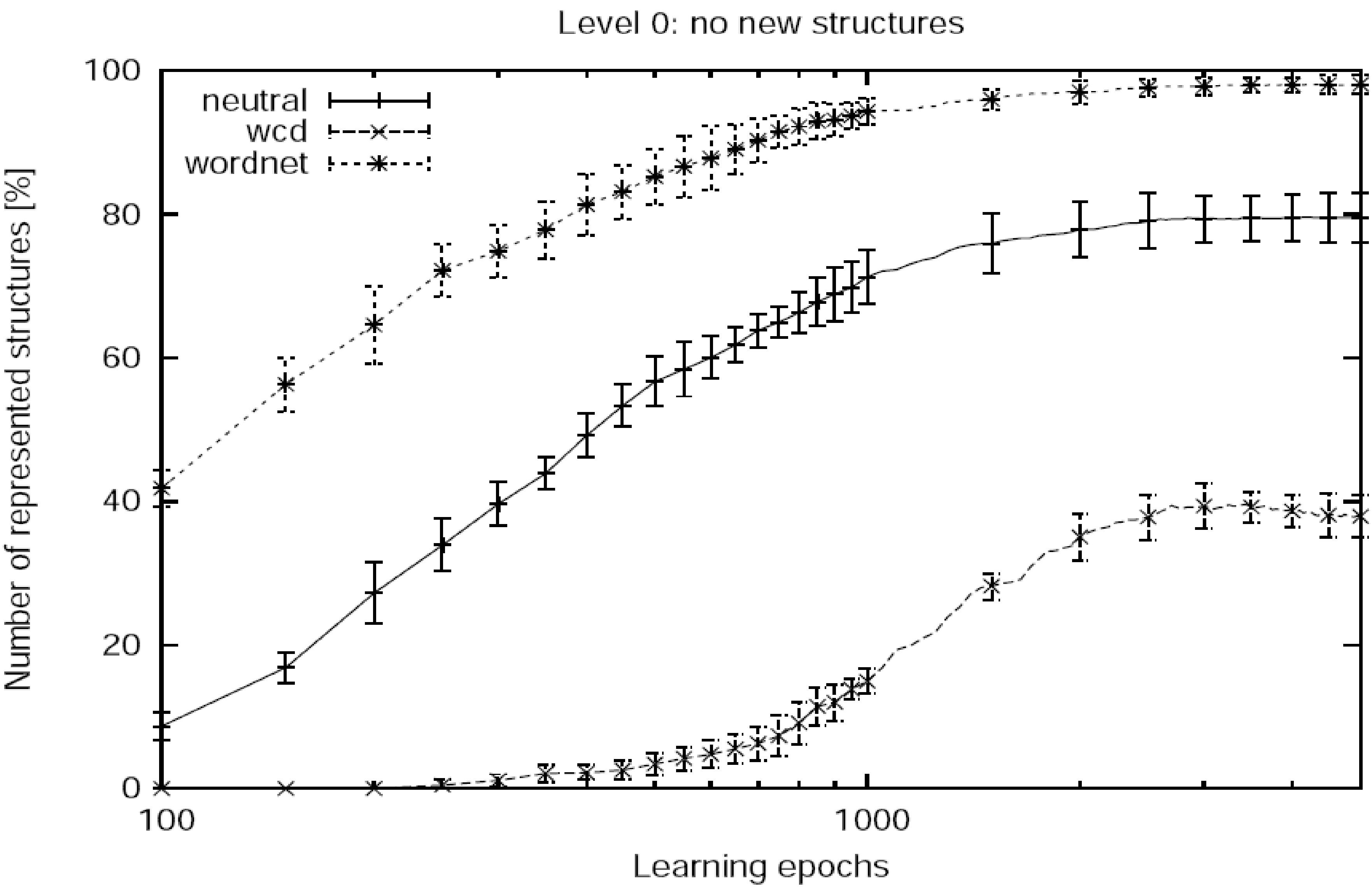
e.g. B in novel object position in testing

3 - Novel atoms (at least one atom never appeared in training)

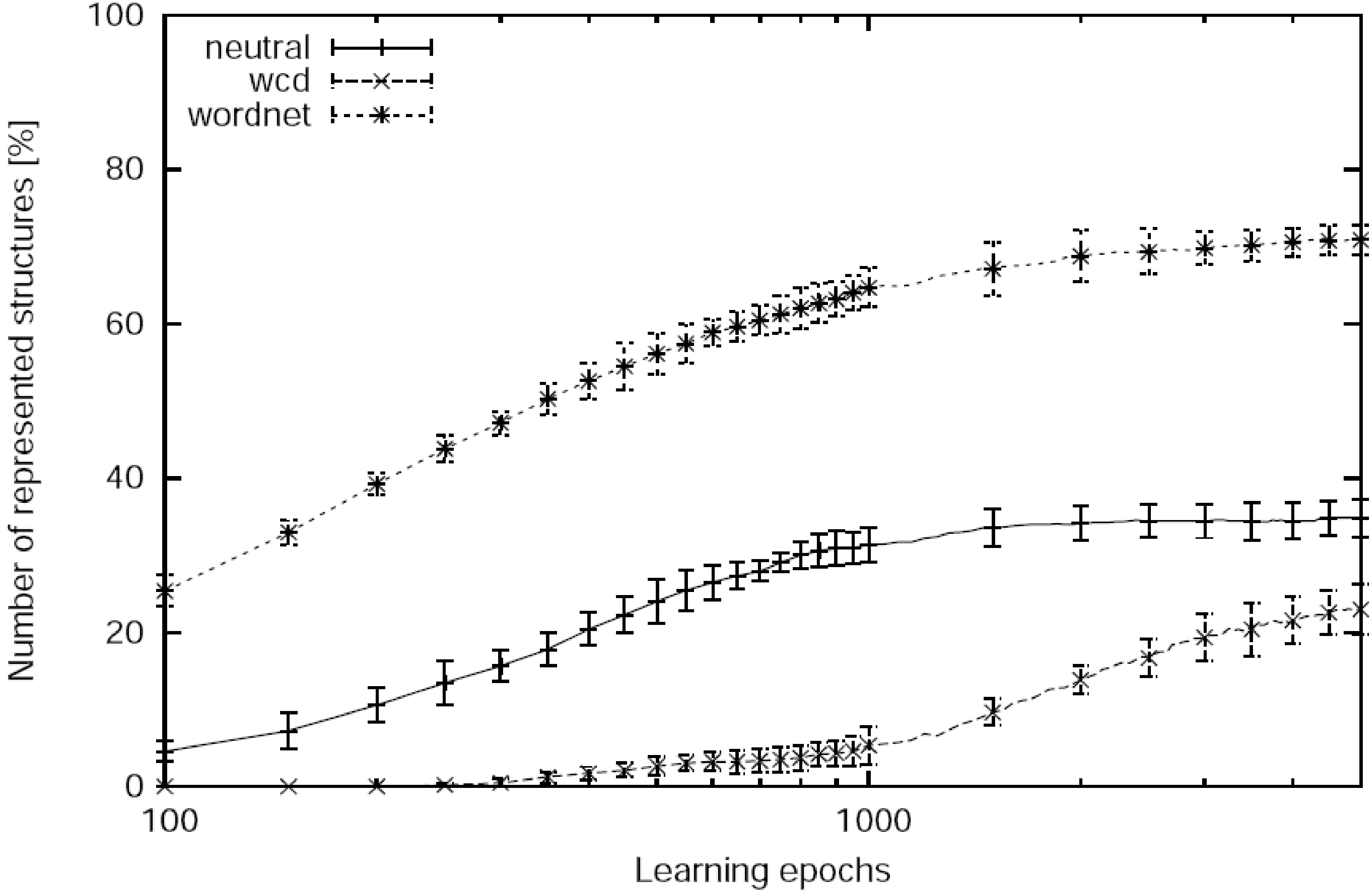
4 - Novel complexity (of test sentences compared to training)

5 - Novel atoms and novel complexity (combination of 3 and 4)

Level 0 - training data

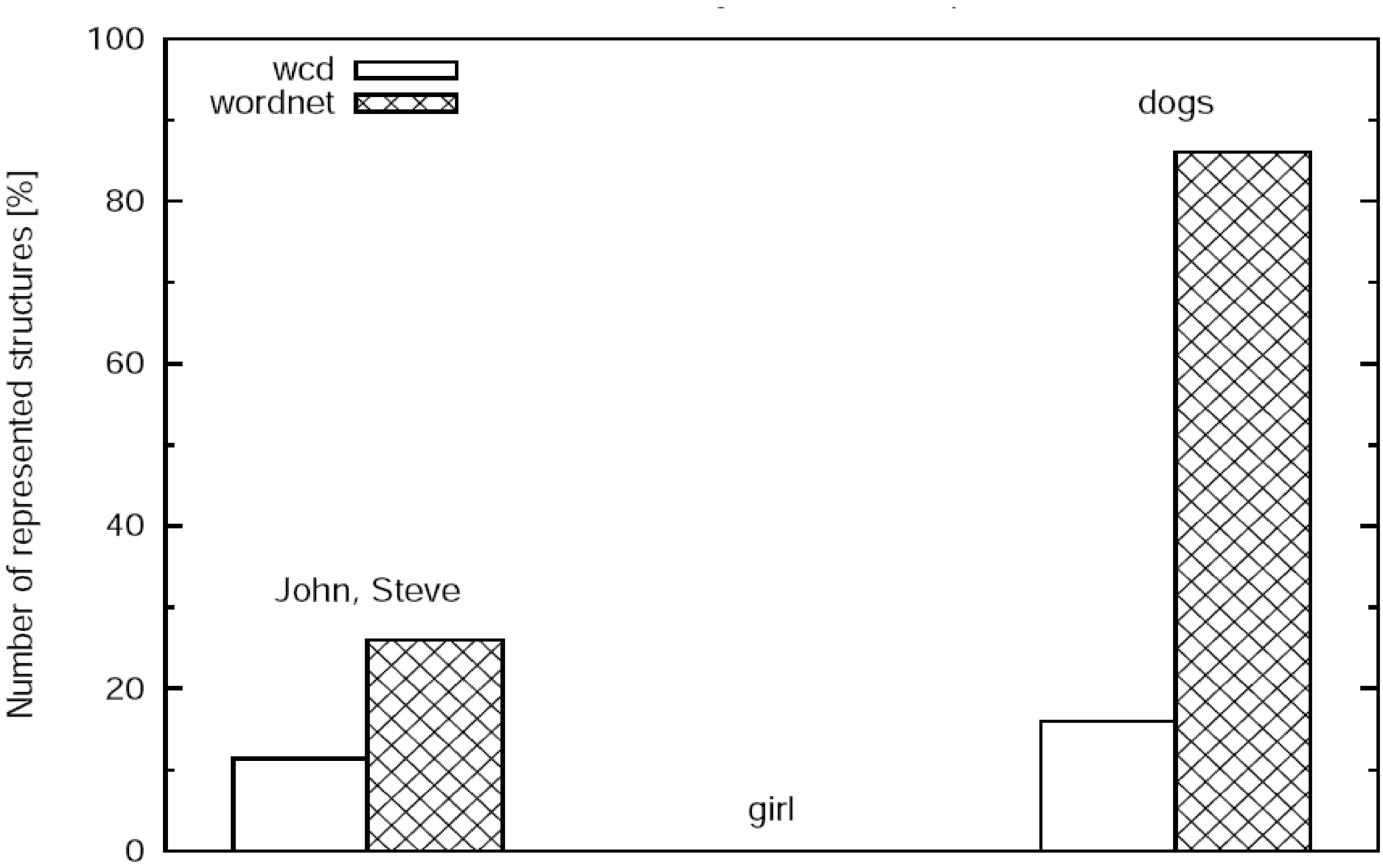


Level 1 – novel sentences



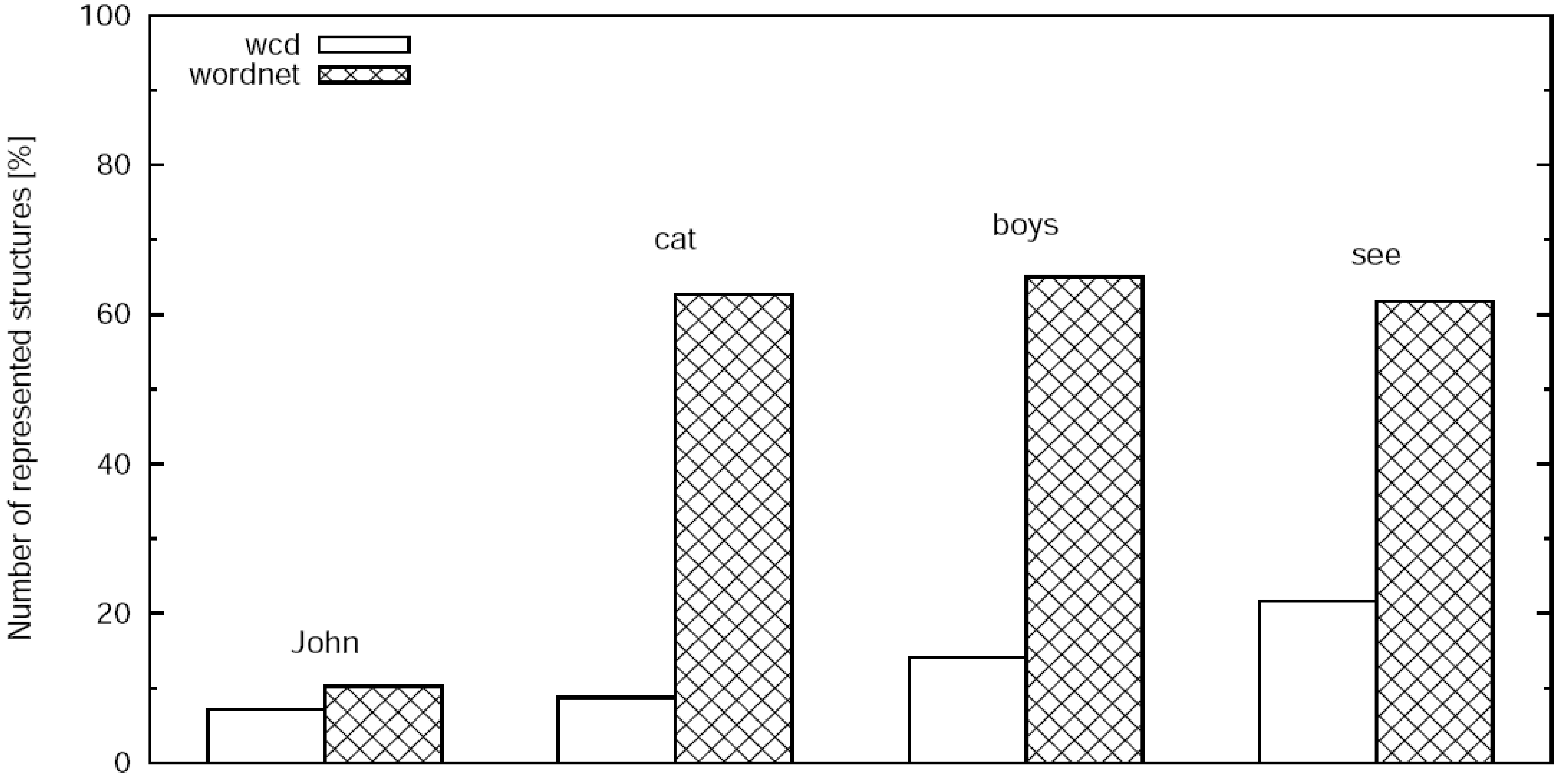
Level 2 – novel terminal positions

- 4 nouns excluded from training as objects



Level 3 – novel atoms in known structures

- 4 data sets, each with one noun completely excluded from training



Summary

linear RAAM confirmed to be superior to RAAM

linear RAAM has some generalization property

benefit of semantic features of words

word co-occurrences were not of much help – confusions in decoding

words with semantic features – some success in testing systematicity (up to level 3)

learning larger trees explicitly with NN may not be cognitively motivating

Ďakujem za pozornosť.