SSST-8, Doha, Qatar 2014.10.24

TRAAM

Learning Bilingual Compositional Distributed Vector Representations of Transduction Grammars

Karteek ADDANKI **Dekai WU**
{ vskaddanki | dekai }@cs.ust.hk

HKUST
Human Language Technology Center
Department of Computer Science and Engineering
University of Science and Technology, Hong Kong

what is **TRAAM**?

Transduction Recursive Auto-Associative Memory

- fully bilingual generalization of monolingual RAAM
- distributed vector representation of SDTGs
- vectors represent **bilingual** constituents
- attractive properties for bilingual grammar induction

modeling recursive structures

- TRAAM **generalizes** from neural network approaches that model **monolingual recursive structures**

- neural language models and SRNs (Bengio *et al.* 2003)
  - contextual history modeled by a RNN

- convolutional networks (Collobert & Weston 2008)
  - learn vector representations of words
  - used in NLP tasks such as POS tagging, chunking and SRL

- recursive auto-associative memory (Pollack 1990)
  - recursive autoencoders are a special case of RAAM:
    RAE successfully applied in sentiment prediction (Socher *et al.* 2011)
  - RAAM is more flexible than convolutional networks:
    URAAM even performs feature structure unification (Stolcke & Wu 1992)

toward TRAAM
bilingual vector space models

- predominantly augment "shake-n-bake" SMT modeling assumptions using feature vectors

- n-gram translation models (Son et al., 2012)
  - bilingual generalization of class based n-grams using distributed representations
  - fails to model compositionality and cross-lingual reordering

- bilingual word embeddings (Zou et al., 2013)
  - recurrent NNLM model with SMT word alignments
  - only learns non-compositional features

toward TRAAM
bilingual vector space models

- NNLMs + input language context (*Devlin et al*. 2014)
  - does not model input and output language features simultaneously
- recurrent probabilistic models (Kalchbrenner & Blunsom 2013)
  - generates an input sentence representation that generates an output sentence
  - lacks structural constraints and relies on a LM to reorder output
- reordering prediction using RAEs (Li e*t al*. 2013)
  - **monolingual** RAEs to predict reordering in a maxent ITG model
  - uses only input language context

why use TRAAM to model bilingual relations?

- compact encoding of subtrees in a constituent
- generalizable representation
- task-dependent representation learning
- elegant recursive use of **both** input and output language features
- feature vector clusters represent **soft categories**

TRAAM
model definition

- uniform feature vector dimension
- compressor network
  - computes feature vector recursively
  - language bias via dimensionality reduction
- reconstructor network
  - generates child vectors and order from parent
  - provides a loss function to drive learning

TRAAM

mathematical formulation

- **bitoken** features are model parameters

- compressor network

$$v_p = \frac{\tanh(W_c[o;v_l;v_r]+b_c)}{\|\tanh(W_c[o;v_l;v_r]+b_c)\|}$$

- reconstructor network

$$[o';v_l';v_r'] = \tanh(W_r v_p + b_r)$$

TRAAM

model training

- initialization
  - bitoken features and $W_c, b_c, W_r, b_r \sim \mathcal{N}(0, \varepsilon)$
- error at each internal node in the biparse
  - linear combination of l2 loss and cross-entropy

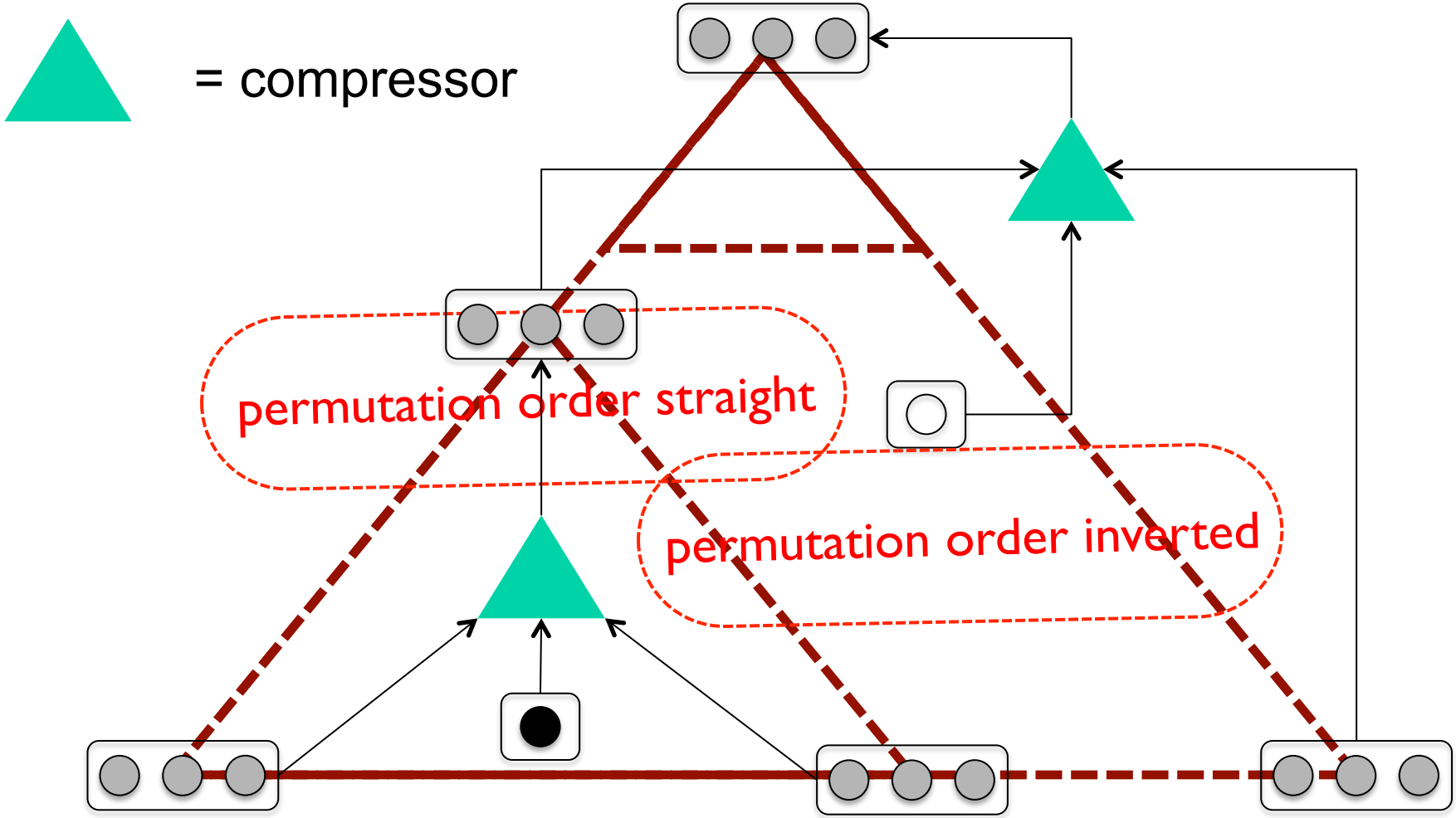
$$E_n = \frac{\alpha}{2} \left| [v_l; v_r] - [v_l'; v_r'] \right|^2 - (1-\alpha)[(1-o)\log(1-o') + (1+o)\log(1+o')]$$

- global loss function with regularization

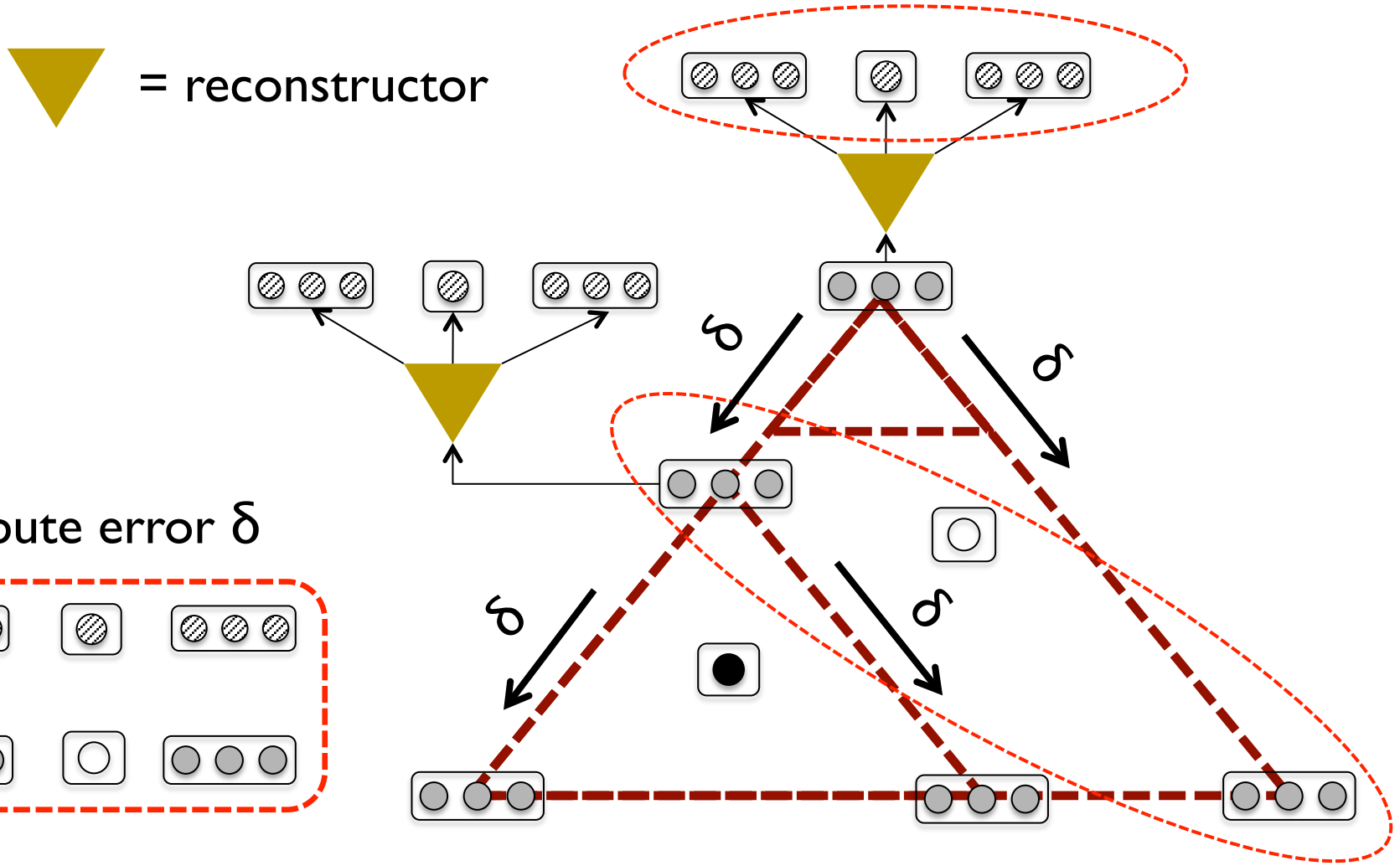
$$J = \frac{1}{T} \Sigma_n E_n + \lambda \left\| \theta \right\|^2$$

- training to minimize loss function

TRAAM forward propagation



TRAAM backpropagation



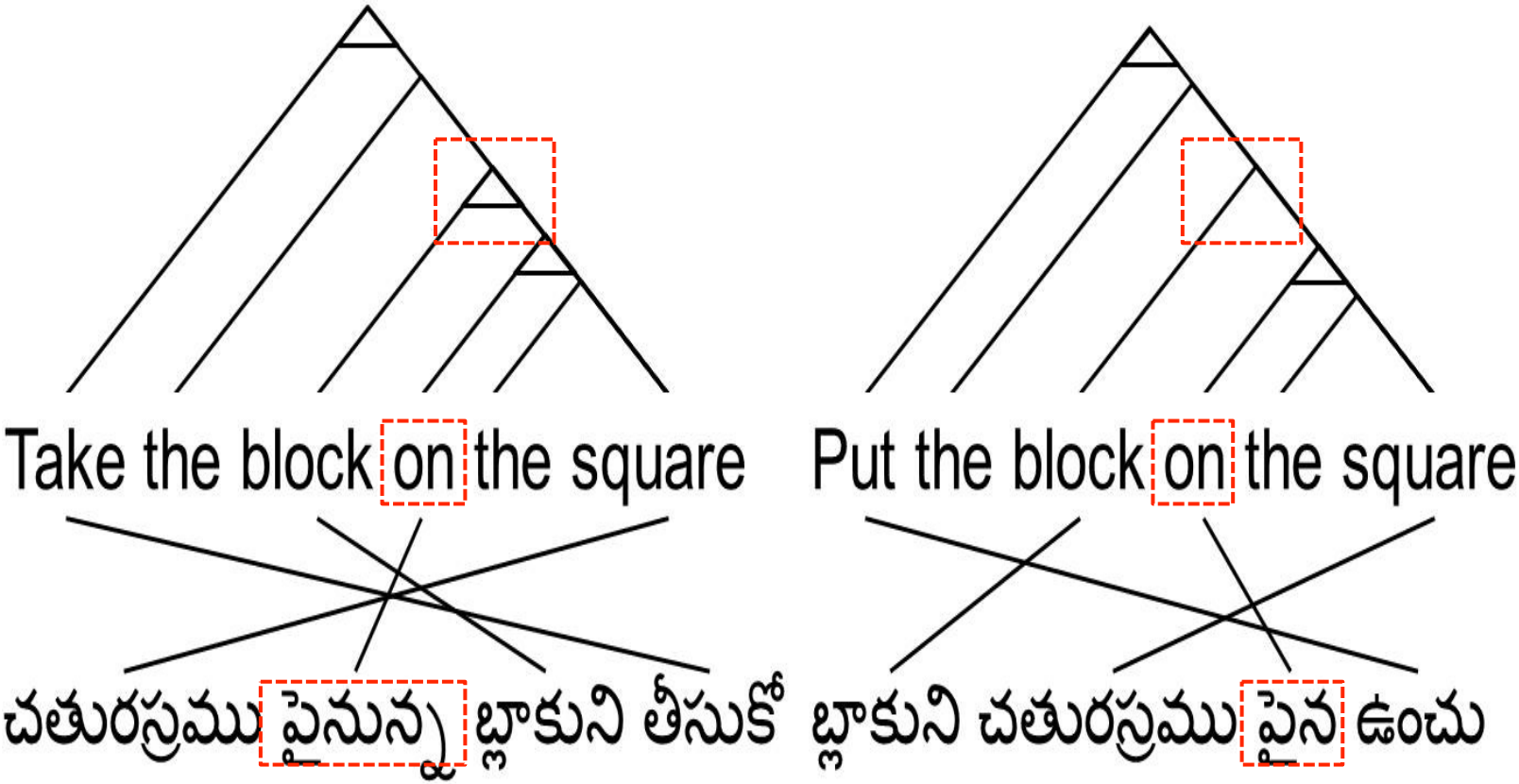
experimental setup

- simple Telugu-English dataset
  - to enable manual inspection of learned features
- Telugu is a Dravidian language with an SOV structure
- blocks world dataset
  - commands to manipulate different colored objects over different shapes
- unlabeled biparses from a unsupervised BITG
  - provide structural constraints

why use TRAAM to model bilingual relations?

- compact encoding of subtrees in a constituent
- generalizable representation
- task-dependent representation learning
- elegant recursive use of **both** input and output language features
- feature vector clusters represent **soft categories**

output language context **matters**!



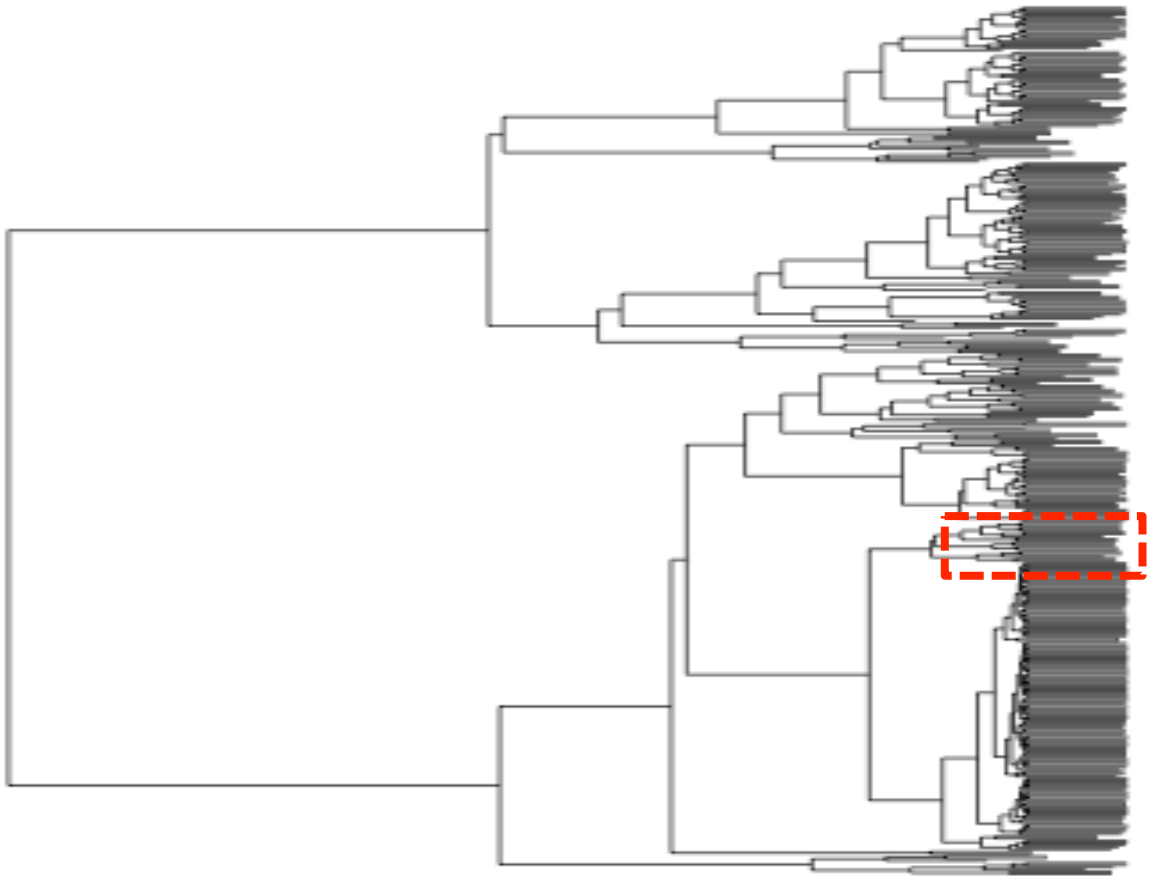
why use TRAAM to model bilingual relations?

- compact encoding of subtrees in a constituent
- generalizable representation
- task-dependent representation learning
- elegant recursive use of **both** input and output language features
- feature vector clusters represent **soft categories**

feature vectors after training

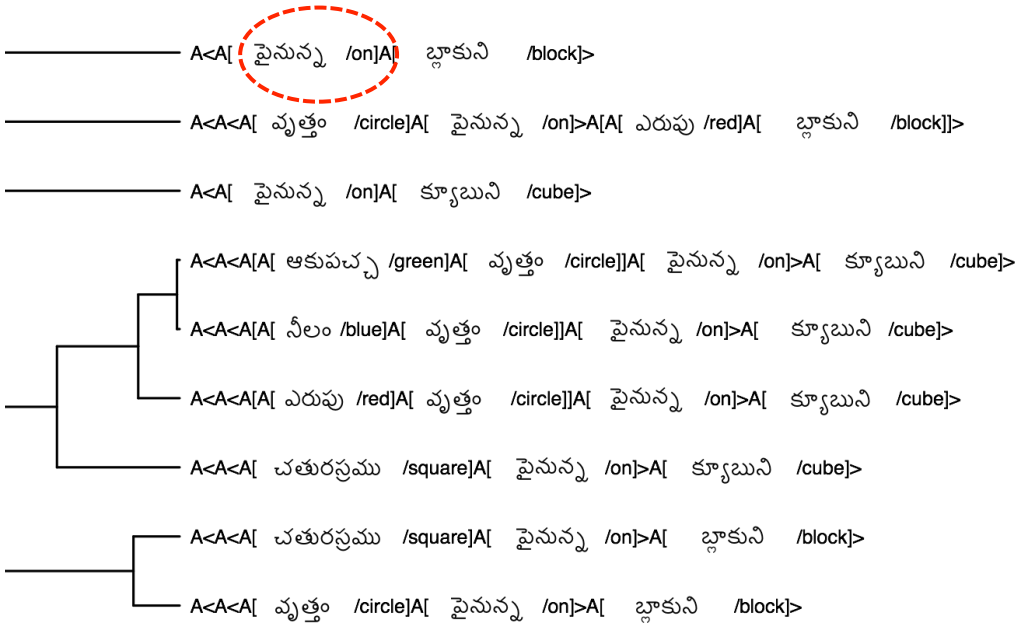
biconstituent	feature vector
A [బ్లాకుని/block]	-0.07,-0.23,-0.07,0.02,-0.06
A[తీసుకో/take]	-0.19,0.03,0.25,0.11,-0.18
A[పైన/on]	0.03,0.09,-0.16,0.09,-0.02
A[పైనున్న/on]	-0.06,0.08,-0.01,0.12,-0.04
A<A[చతురస్రము/square]A[పైన/on]>	0.77,0.61,-0.15,0.88,-0.60
A<A[వృత్తం/circle]A[పైన/on]>	0.82,0.51,-0.12,0.70,-0.47
A<A[చతురస్రము/square]A<A<A[పైనున్న/on]A[బ్లాకుని/block]>A[తీసుకో/take]>>	0.59,0.57,0.02,0.91,-0.62

dendrogram of biconstituent feature vectors



zooming in
dendrogram of biconstituent feature vectors

- fuzzy category
- describe an object wrt position on another object
- single sense of “on” (పైనున్న)
- other clusters reveal such similarites



conclusions

- **TRAAM** Transduction Recursive Auto-Associative Memory
- fully bilingual generalization of monolingual RAAM
  - can model arbitrary rank SDTGs
- feature vector specifies a relation between
  - two monolingual constituents
  - permutation order
- sensitive to *both* input and output language contexts
  - vectors represent bilingual instead of monolingual similarities
  - attractive for inducing differentiated bilingual categories
- worth detailed exploration!