Knowledge Processing with Neural Networks

Lecture 7: Principles of Hybrid Neural Symbolic Architectures

http://www.informatik.uni-hamburg.de/WTM/

Example for cognition?
Intelligent behaviour on a cow?

Motivation for hybrid processing

- The brain is a hybrid system supporting different forms of processing: signals, symbols, structures, knowledge
- Hybrid processing combines or integrates multiple modes of processing
- Hybrid systems in artificial intelligence and knowledge engineering for increasing performance
- Hybrid processing in cognitive science and cognitive neuroscience for plausible cognitive models

Some examples for hybrid integration

- Cognition is not precise! (“go close to the table”)
- Integration of hybrid forms of processing
  - multimodal integration of cognitive processing
  - speech / language integration (signals and concepts…)
  - information retrieval (text and images…)

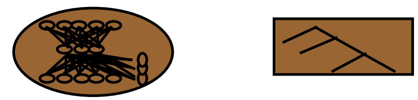
motivate

- Integration of hybrid methods
  - integration of neural and symbolic techniques
  - combination of symbolic fuzzy and statistical methods…

Framework of different types of hybrid integration architectures

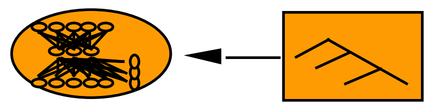
Hybrid transfer architectures: knowledge is transferred to abstract level

Symbolic representation
of neural representation



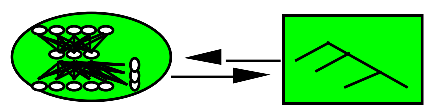
Hybrid processing architectures for hybrid realization of symbolic structures

Loose coupling



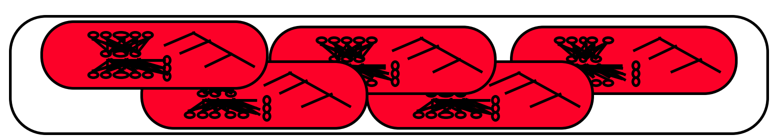
Symbolic and neural modules separate
and unidirectional communication

Tight coupling



Symbolic and neural modules
separate and bidirectional communication

Integration

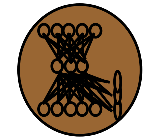


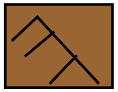
Symbolic and connectionist modules fully
embedded and integrated

Hybrid transfer architectures

- Symbolic representation of neural representation
- Automatic insertion or extraction
- Advantages for choosing good architectures
- Explanation of neural networks
- Induction of well-known knowledge

Weight-based transfer

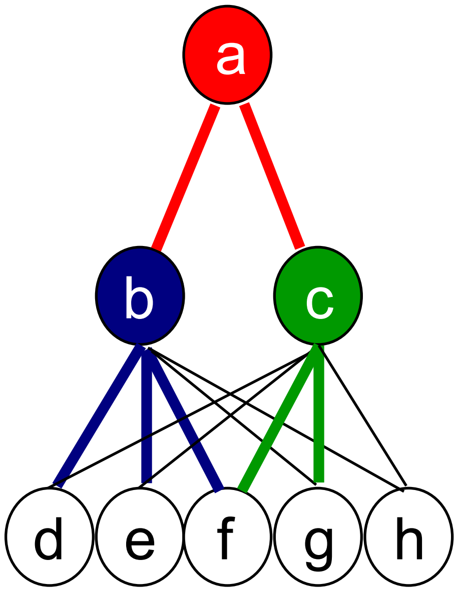




- Connectionist weights contain the knowledge
- Problem: distributed representations are difficult to understand and modify
- Transfer of weights into symbolic rules
- Often used for feedforward networks
- Grouping, elimination and clustering of weights
- N-of-M rules

Rule extraction in a hybrid transfer architecture

Feedforward network



Symbolic rules

a :- b , c

b :- d , e , f

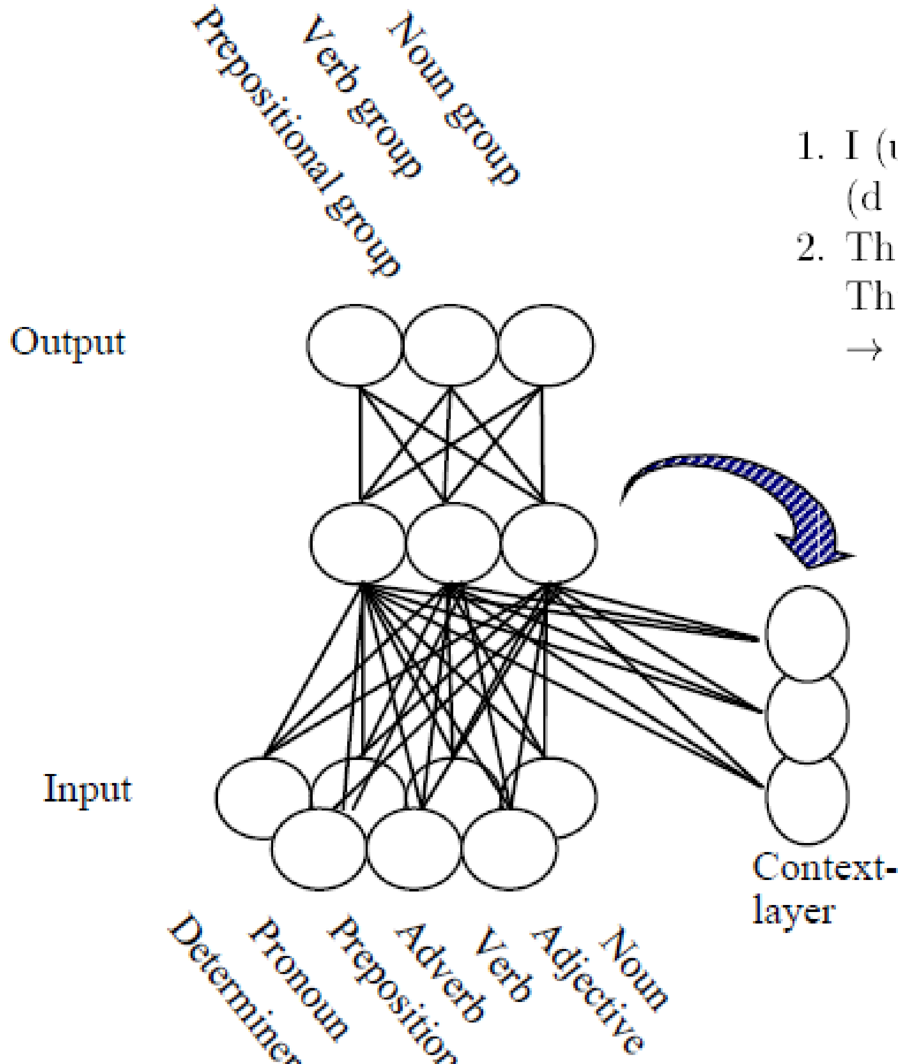
c :- f , g

Knowledge Extraction from Transducer Networks

- Example from a recurrent transducer network for syntactic phrase assignment
- What is trained in examples and what is to be learned is the mapping between basic and more abstract syntactic categories as shown here:

1. I (u $\rightarrow$ ng) thought (v $\rightarrow$ vg) in (r $\rightarrow$ pg) the (d $\rightarrow$ pg) next (j $\rightarrow$ pg) week (n $\rightarrow$ pg)
2. That (u $\rightarrow$ ng) is (v $\rightarrow$ vg) the (d $\rightarrow$ ng) Thursday (n $\rightarrow$ ng) after (r $\rightarrow$ pg) Easter (n $\rightarrow$ pg)

Knowledge Extraction from Transducer Networks



1. I (u → ng) thought (v → vg) in (r → pg) the (d → pg) next (j → pg) week (n → pg)
2. That (u → ng) is (v → vg) the (d → ng) Thursday (n → ng) after (r → pg) Easter (n → pg)

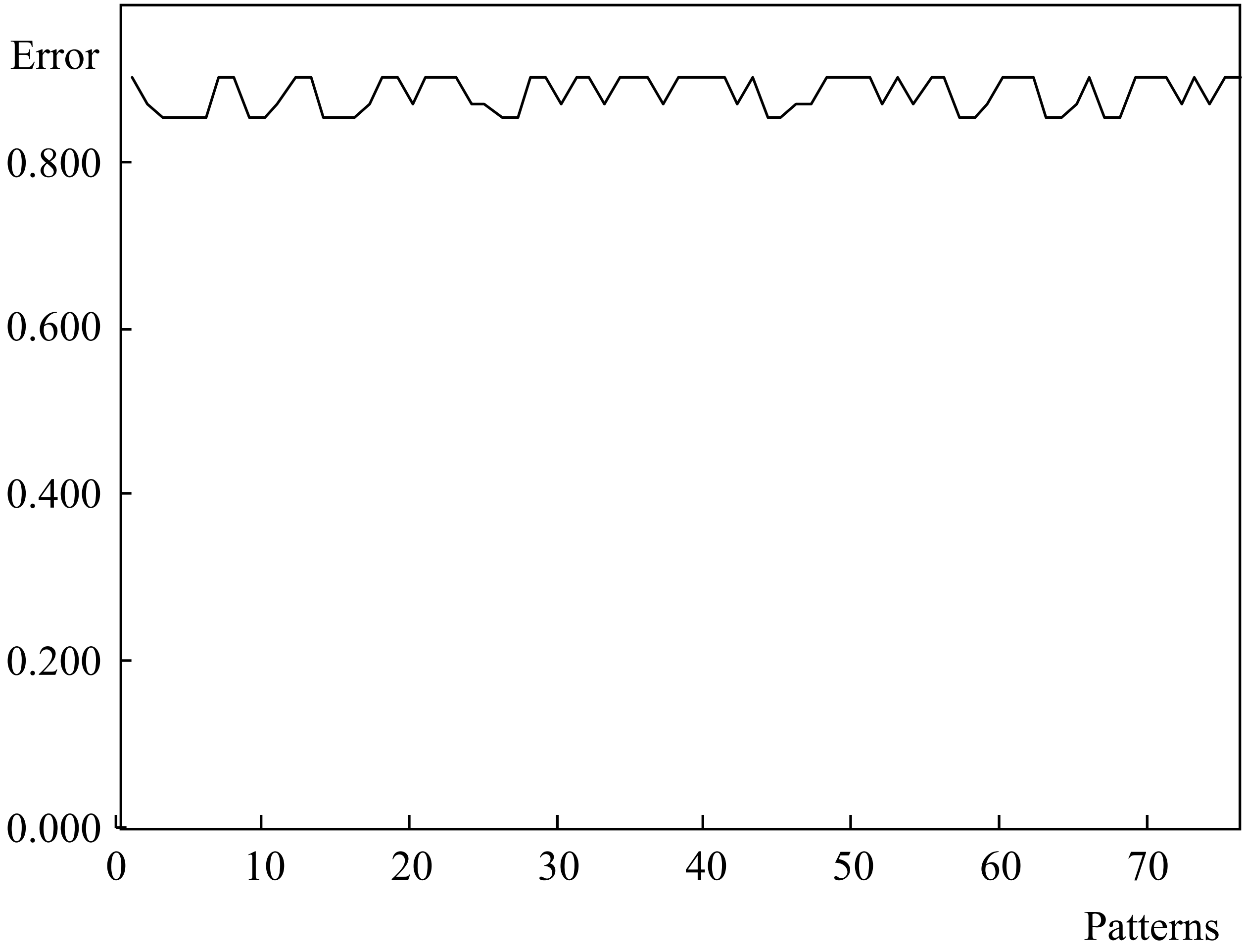
How to interpret recurrent networks?

- Cluster analysis, activation values, weight interpretation e.g. Hinton Diagrams
- Stepwise dynamic analysis of the learning behavior over time
- Symbolic interpretation of neural networks as transducers
- For simple demonstration and interpretation we consider a simple SRN architecture: 7 input of syntactic classes, 3 output units, 76 patterns

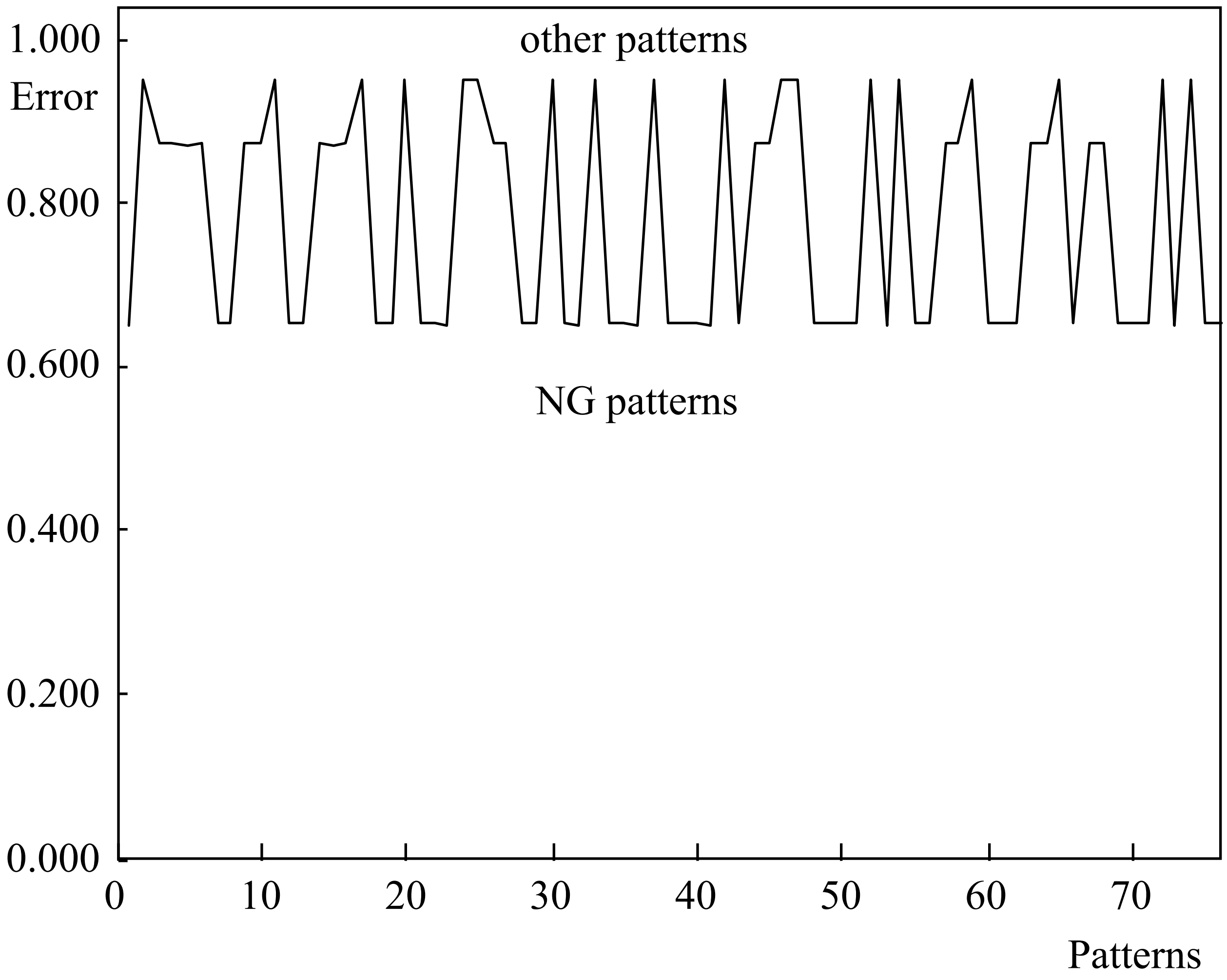
Stepwise dynamic analysis of the learning behavior

- Lazy learning strategy
- Initial high error equally distributed
- Learning of often occurring noun groups
- Learning of prepositional groups, verb groups later
- Learning of sequential context later
- Learning of exceptions

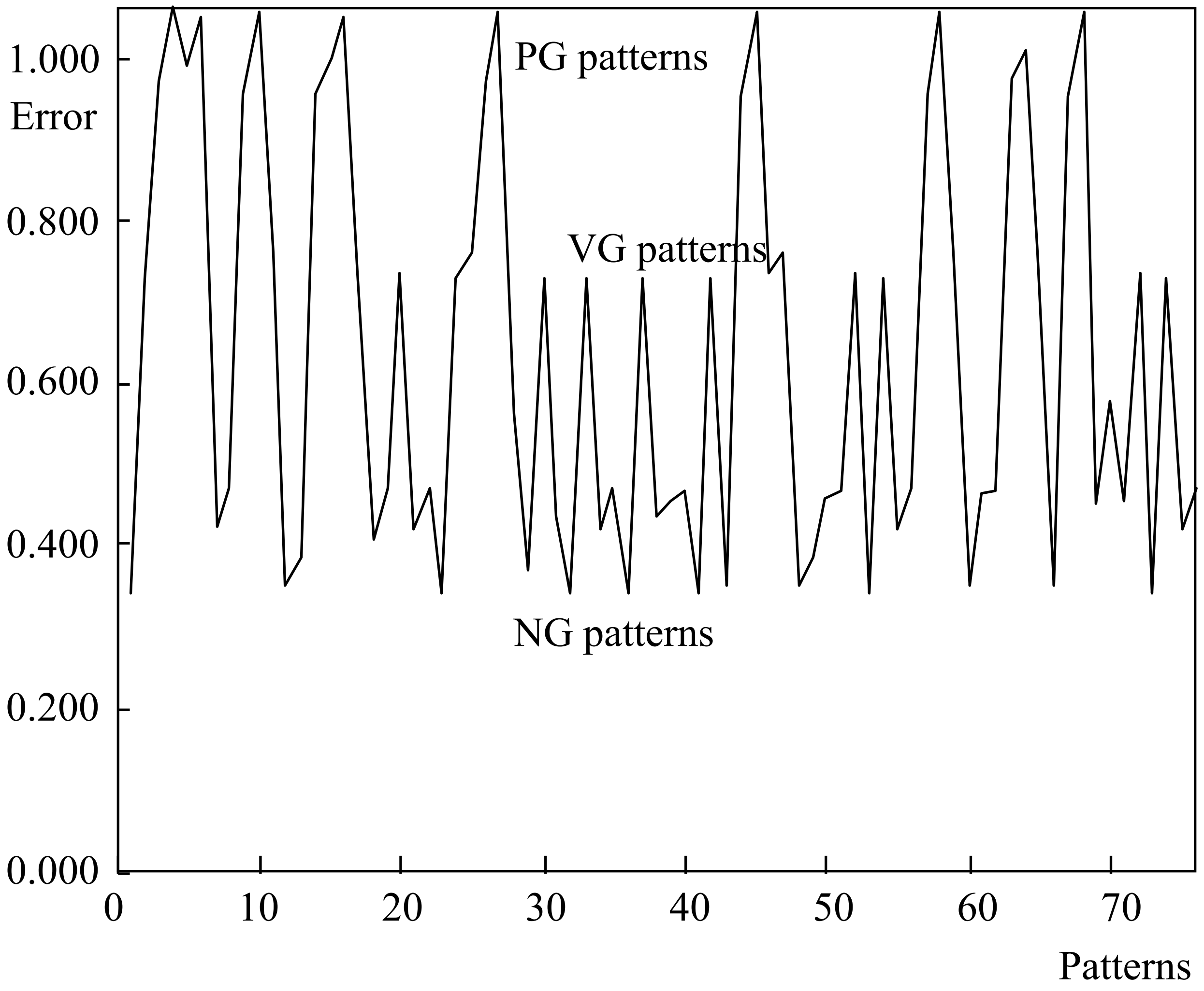
Patterns before learning



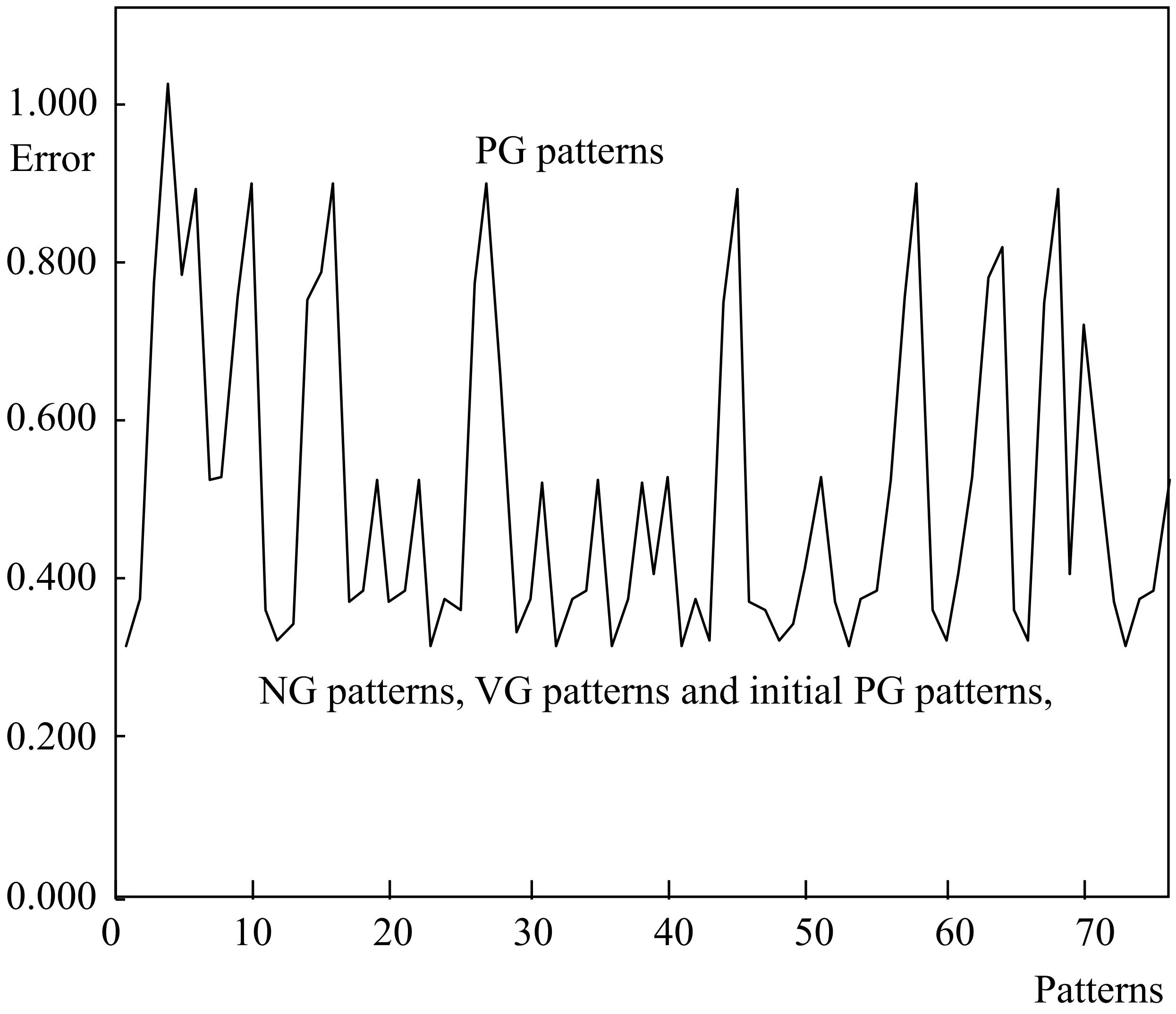
After 100 patterns



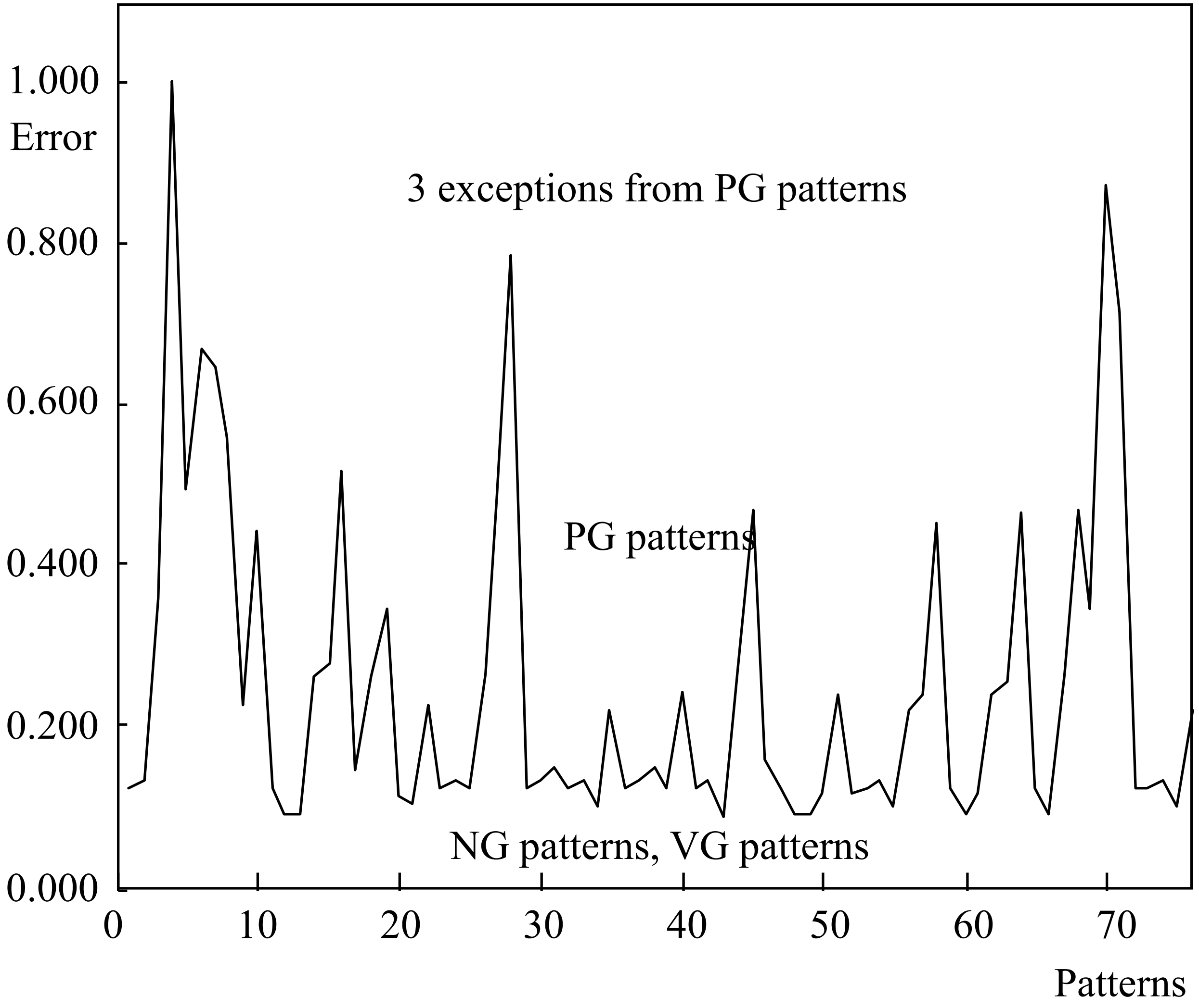
After 600 patterns



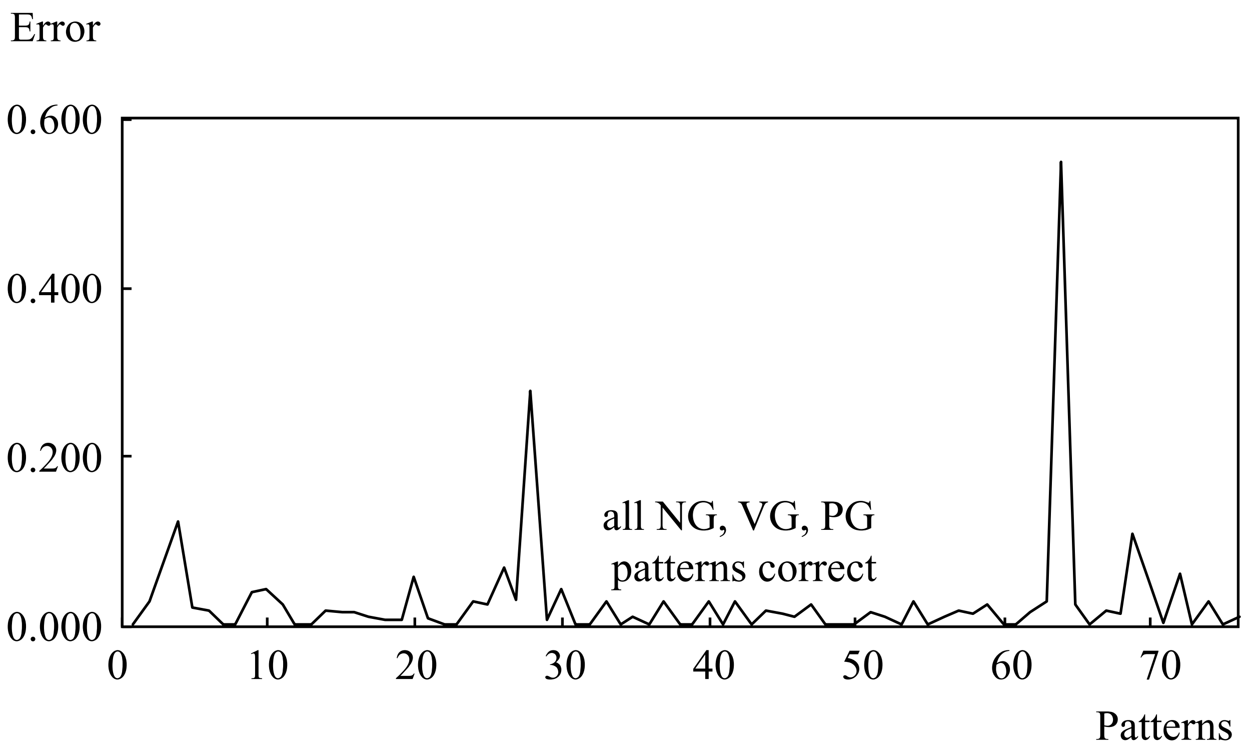
After 1400 patterns



After 3000 patterns

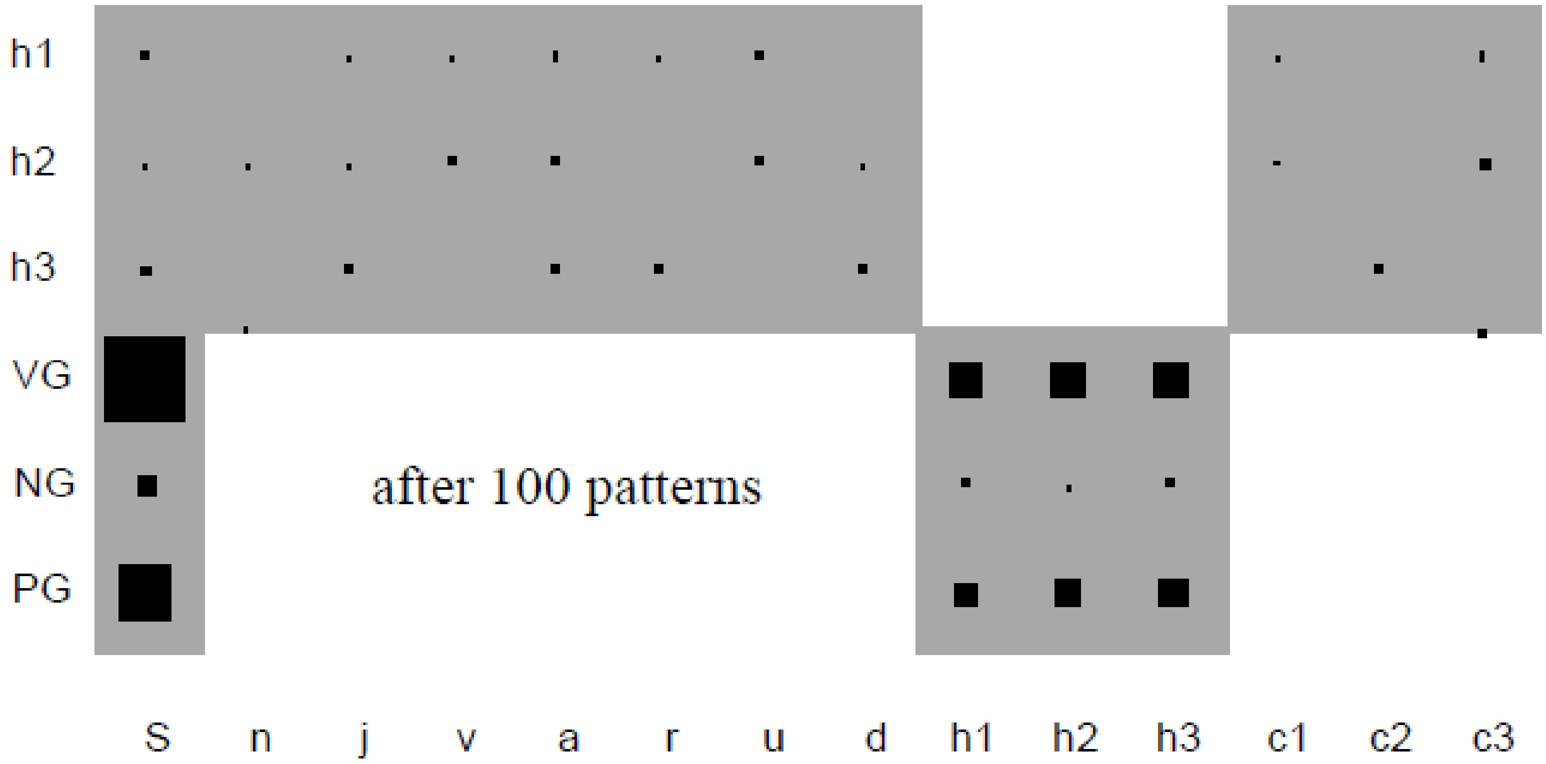


After 150000 patterns



Weight analysis for knowledge extraction (1)

- Identifiers of the source “neurons” shown horizontally and identifiers of the goal neurons vertically.
- White represent positive, black boxes negative weights

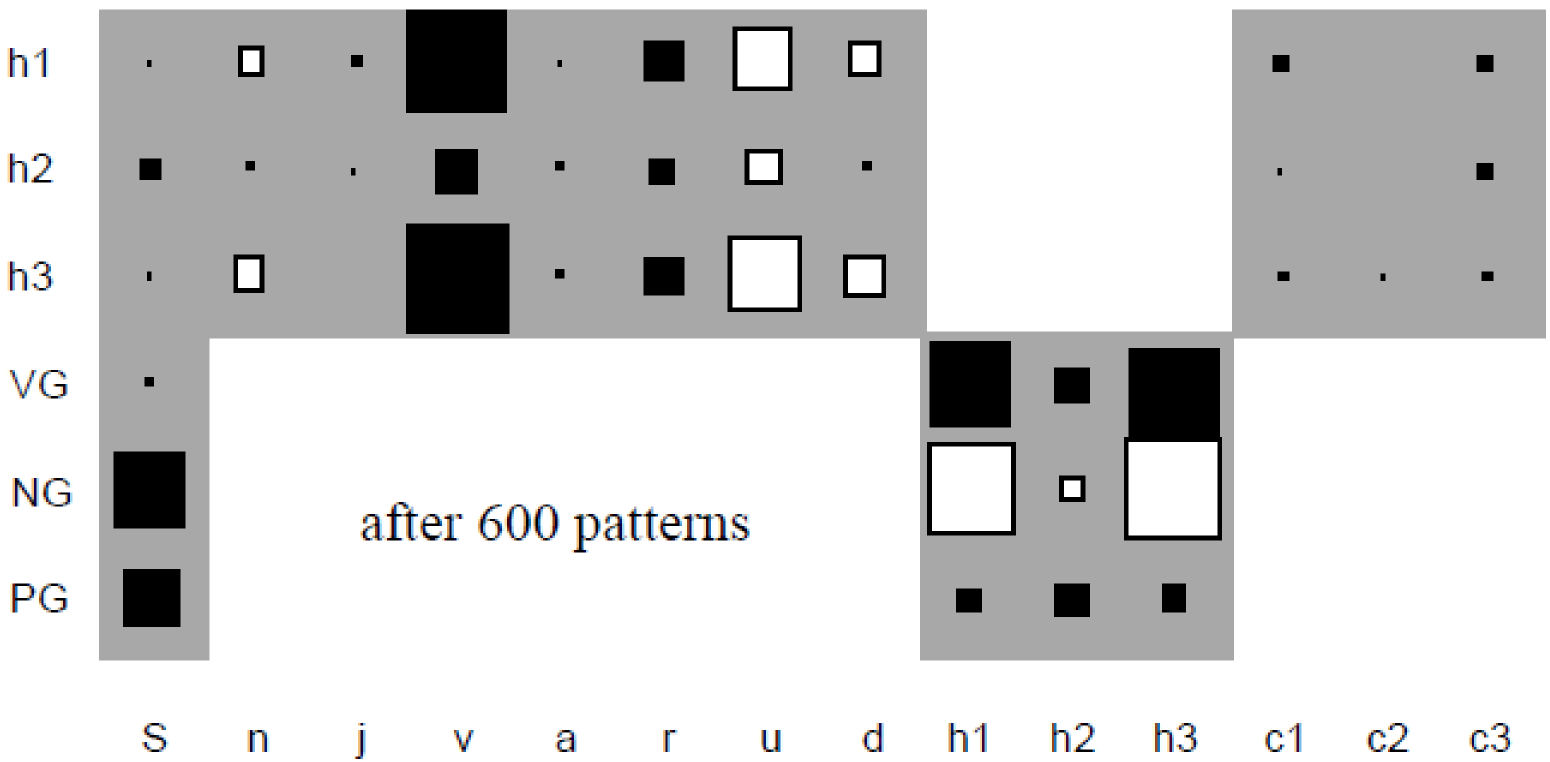


Weight analysis for knowledge extraction (2)

- From left to right, we can see the weights from the bias neuron (*S*), from neurons for the syntactic basic categories (*n, j, v, a, r, u, d*), from the three internal neurons (*h1, h2, h3*) and from the three context neurons (*c1, c2, c3*).
- In the vertical axis from top to bottom, we see the weights to the three internal elements (*h1, h2, h3*) and to the output elements representing the abstract syntactic categories (VG, NG, PG).
- Over time the weights adapt and change

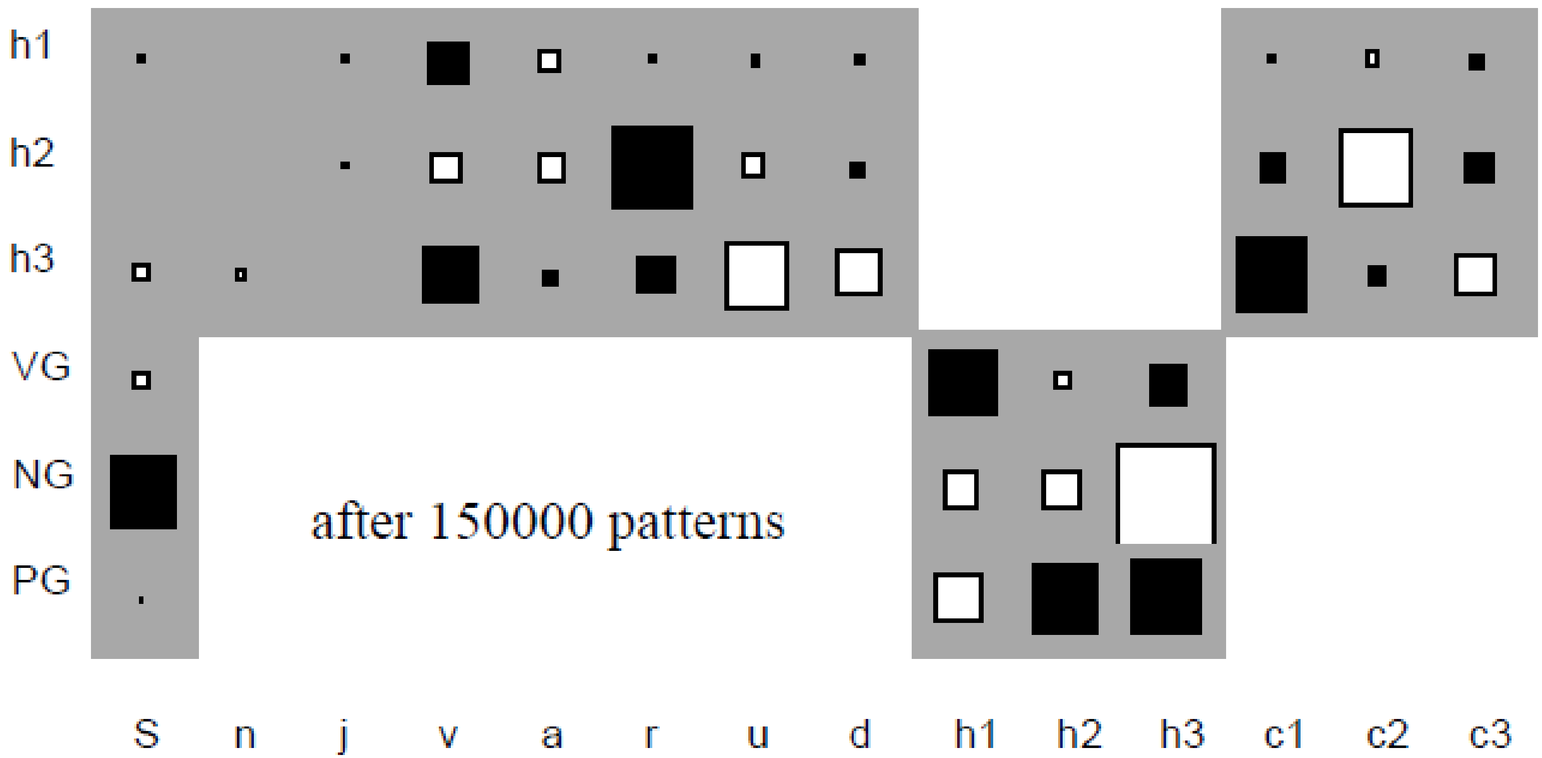
Weight analysis for knowledge extraction (3)

- White boxes represent positive weights, black boxes negative weights; NG and VG patterns learned



Weight analysis for knowledge extraction (4)

- White boxes represent positive weights, black boxes negative weights; NG, VG, VP patterns learned

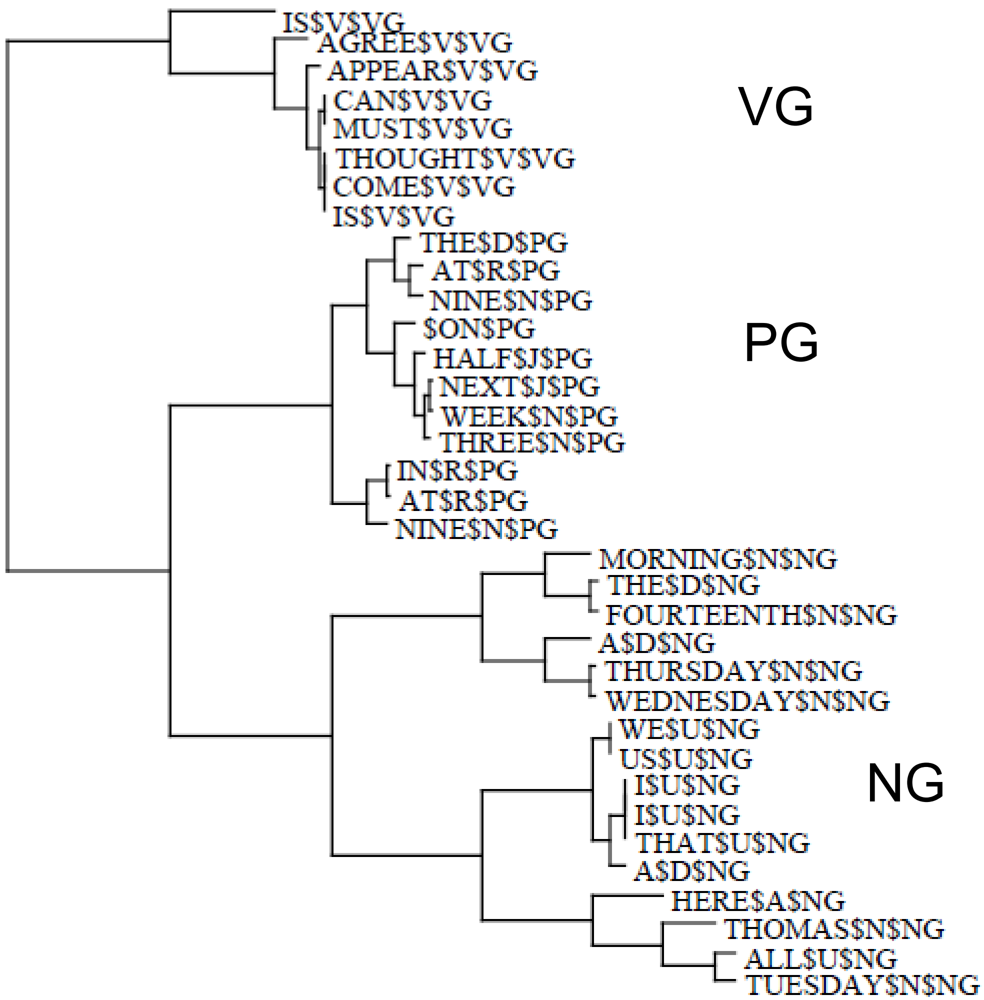


Weight analysis for knowledge extraction (5)

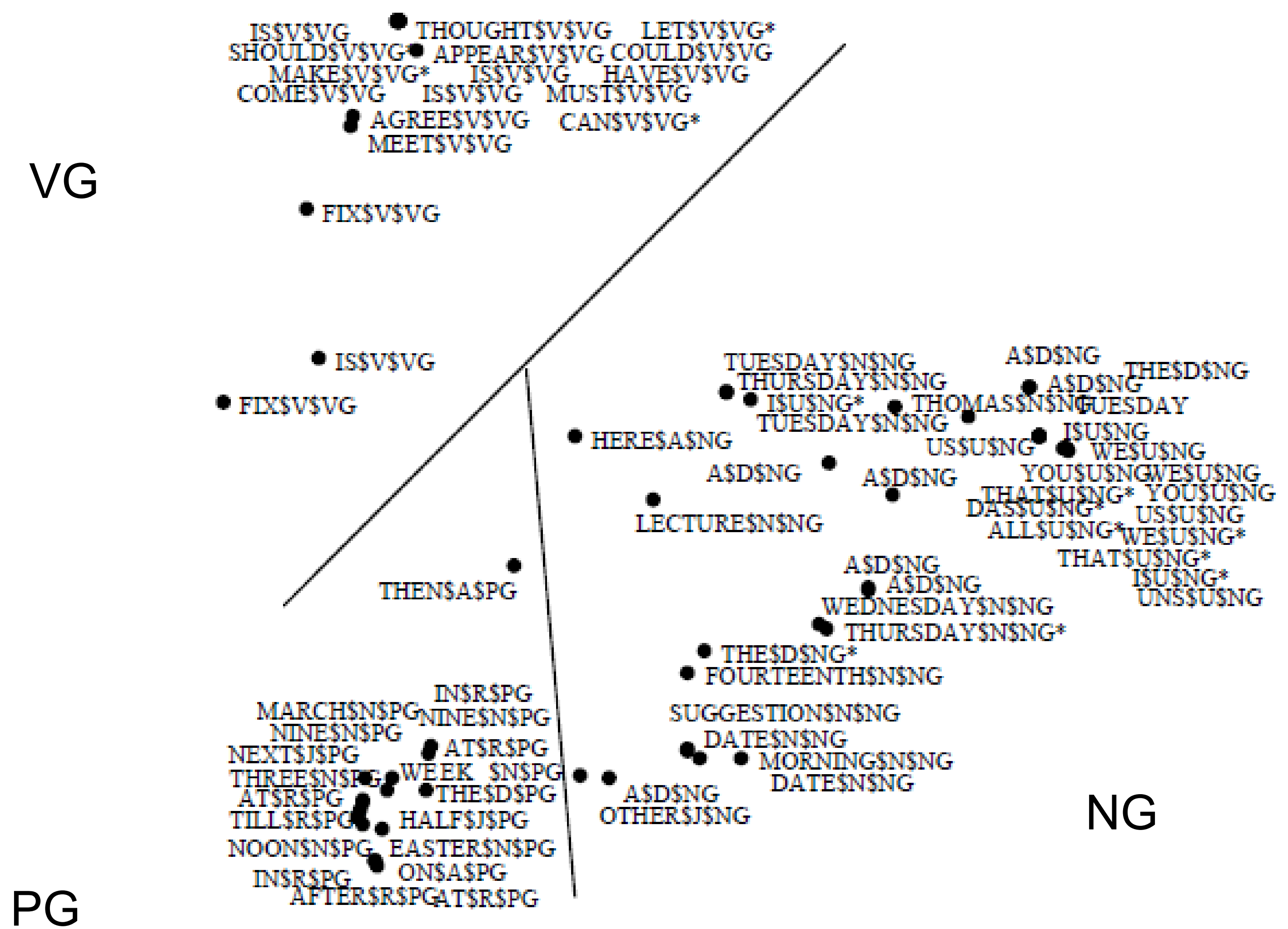
- We can explain certain phenomena with weight analysis
- However difficult to extract ***explicit*** knowledge
- Static representation of weights does not show ***dynamics*** of recurrent network
- Distribution of weights difficult to show for larger networks

Activation analysis for knowledge extraction

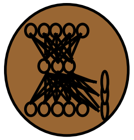
- We can cluster the activation patterns of all internal representations of sequences
- The internal layer represents the learned knowledge about abstract categories as shown in the dendrogram

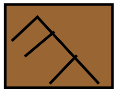


Activation analysis using principal component analysis



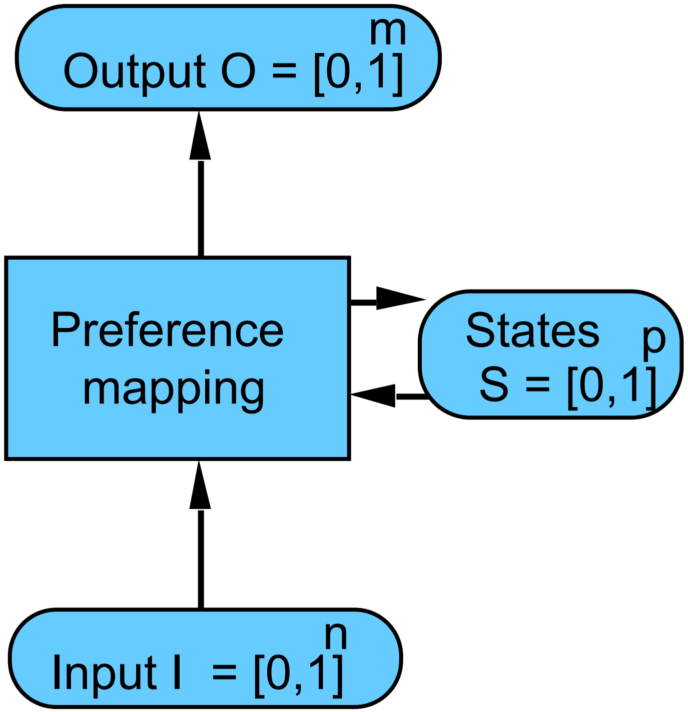
Activation-based automata extraction

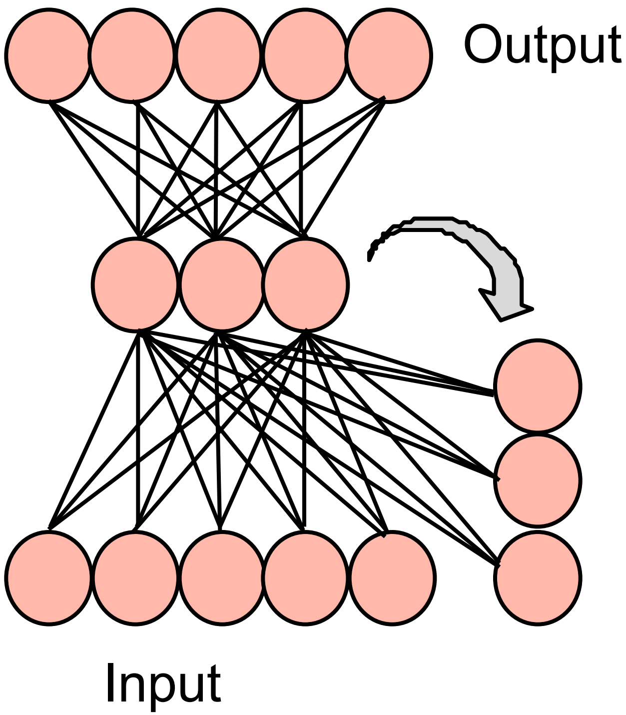




- Weights represent ***knowledge*** at a very detailed level
- Activation values represent more ***integrated knowledge*** for particular patterns
- Activations of internal representations can be used for automata ***extraction***
- Recurrent networks transferred to finite state automaton
- Typical symbolic knowledge:
  - if state is *x* and input is *y* then go to state *z*

Preference Moore Machines and recurrent neural networks



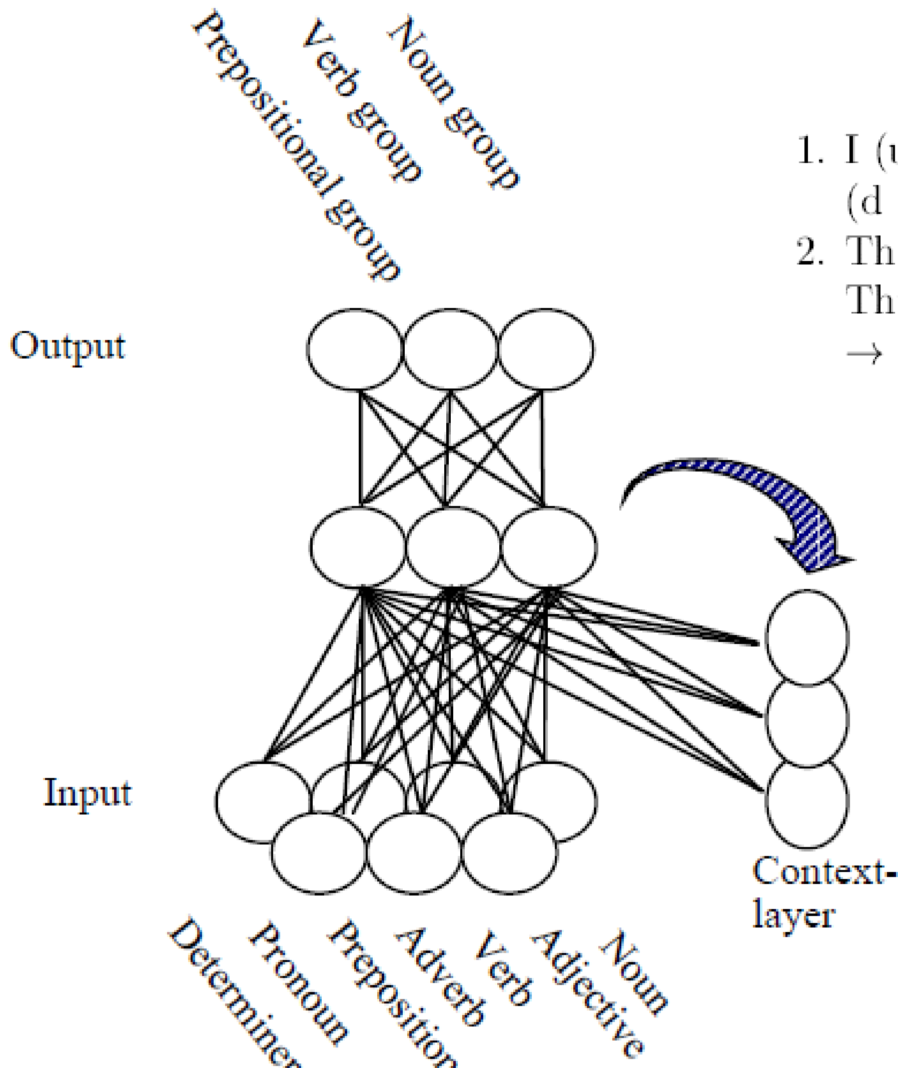


if state is *a* and input is *b* then go to state *x* and output is *y*

Relationship of automata and recurrent neural networks

- Weights and activations so far represented statically
- Dynamics of recurrent connections not well represented
- Recurrent networks transferred to finite state automaton
- For each output vector and each concatenation of input and state vector find the closest symbolic representations (closest corner preference vector according to some metric, e.g. Euclidean)

Knowledge extraction from transducer networks

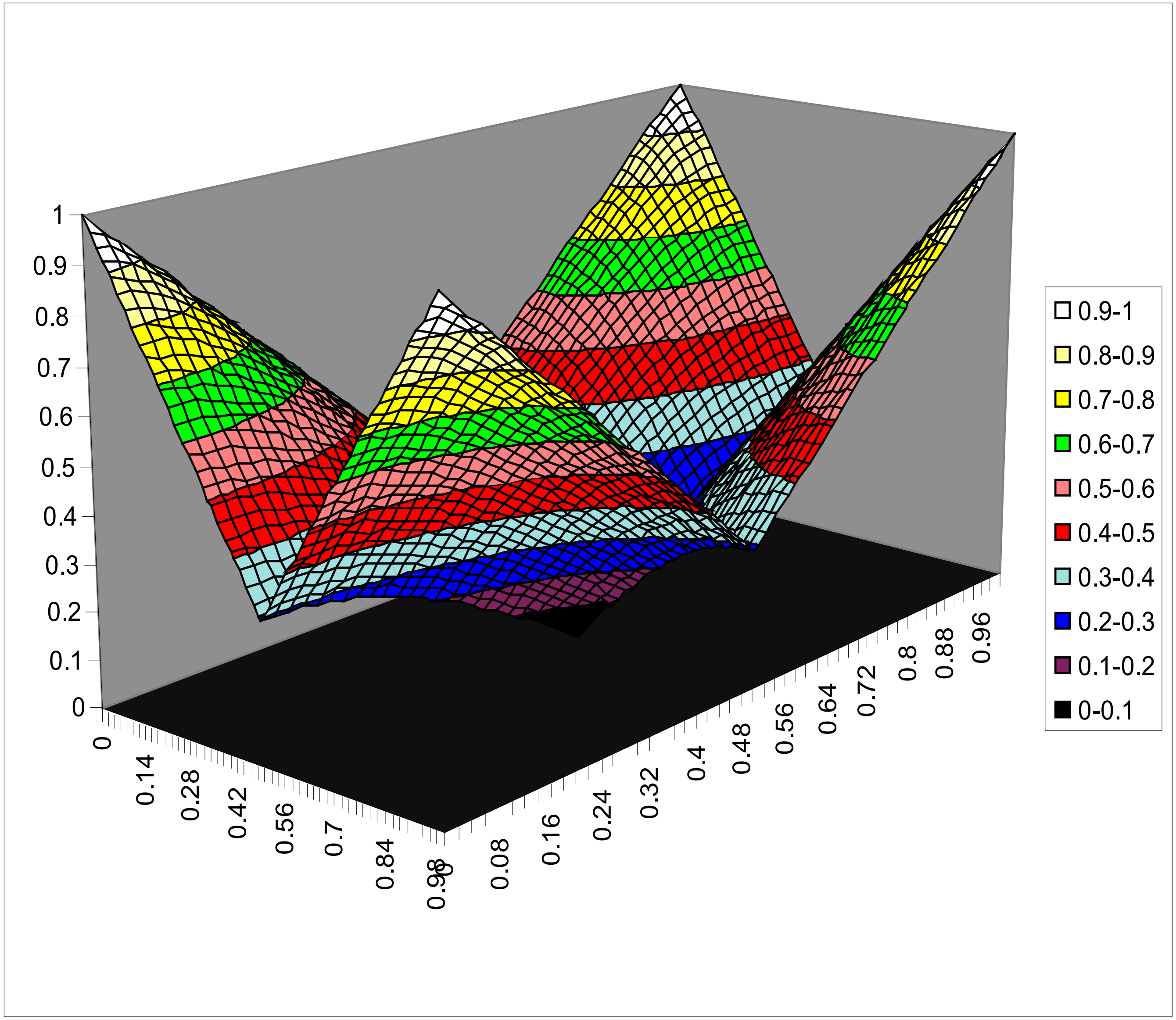


1. I (u $\rightarrow$ ng) thought (v $\rightarrow$ vg) in (r $\rightarrow$ pg) the (d $\rightarrow$ pg) next (j $\rightarrow$ pg) week (n $\rightarrow$ pg)
2. That (u $\rightarrow$ ng) is (v $\rightarrow$ vg) the (d $\rightarrow$ ng) Thursday (n $\rightarrow$ ng) after (r $\rightarrow$ pg) Easter (n $\rightarrow$ pg)

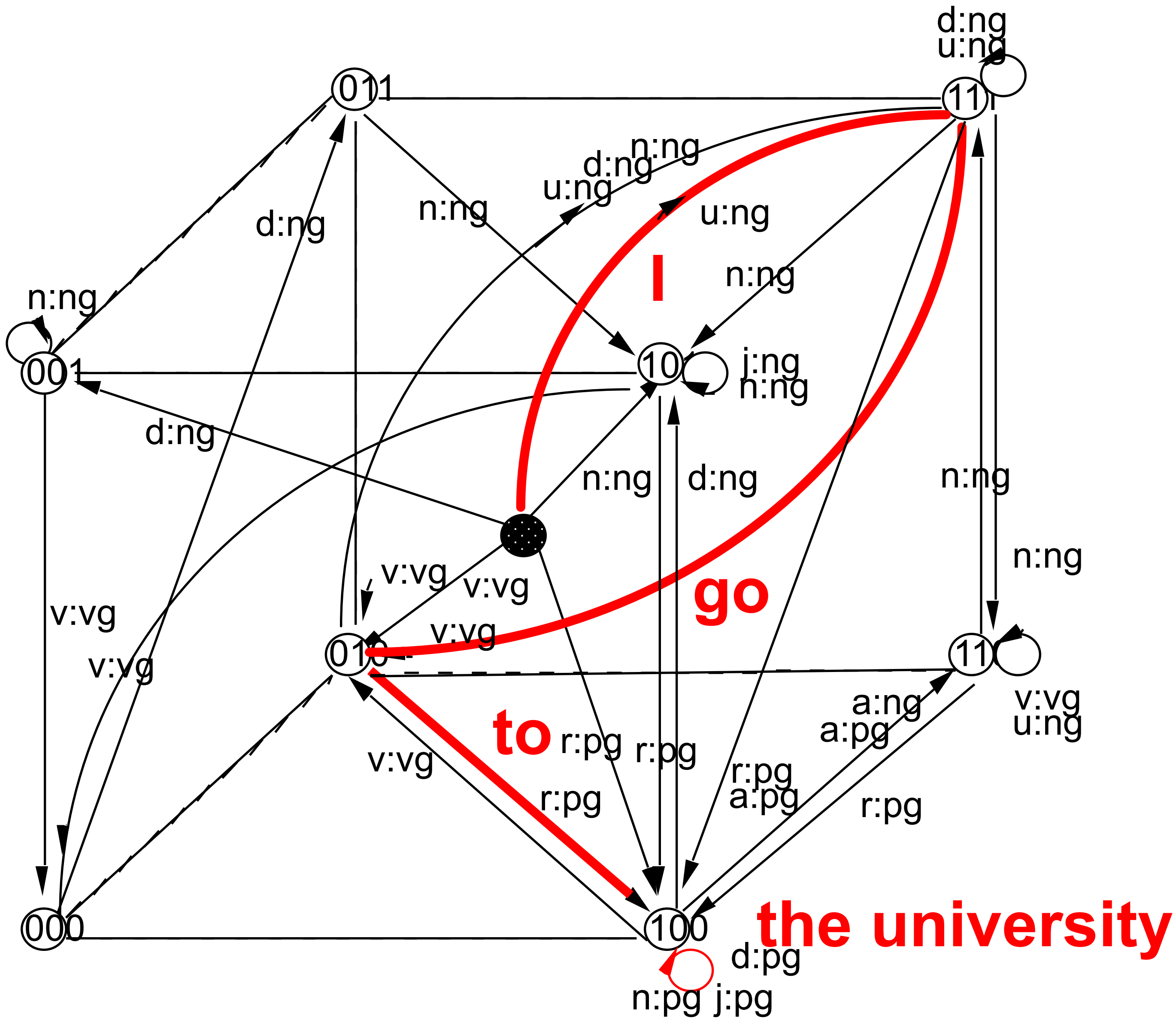
Symbolic interpretation of recurrent networks as transducers

- Need to have a mapping from arbitrary vectors to symbolic representations
- Presentation of all patterns
- Computation of ***next corner preference*** for each output vector and state vector
- Computation of symbolic transducer
- Flexibility of abstraction level

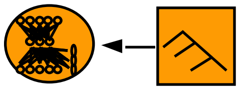
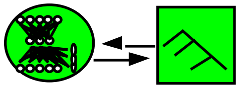
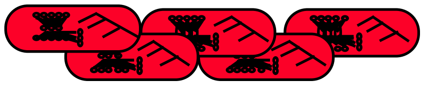
Next corner preferences in 2-dimensional space



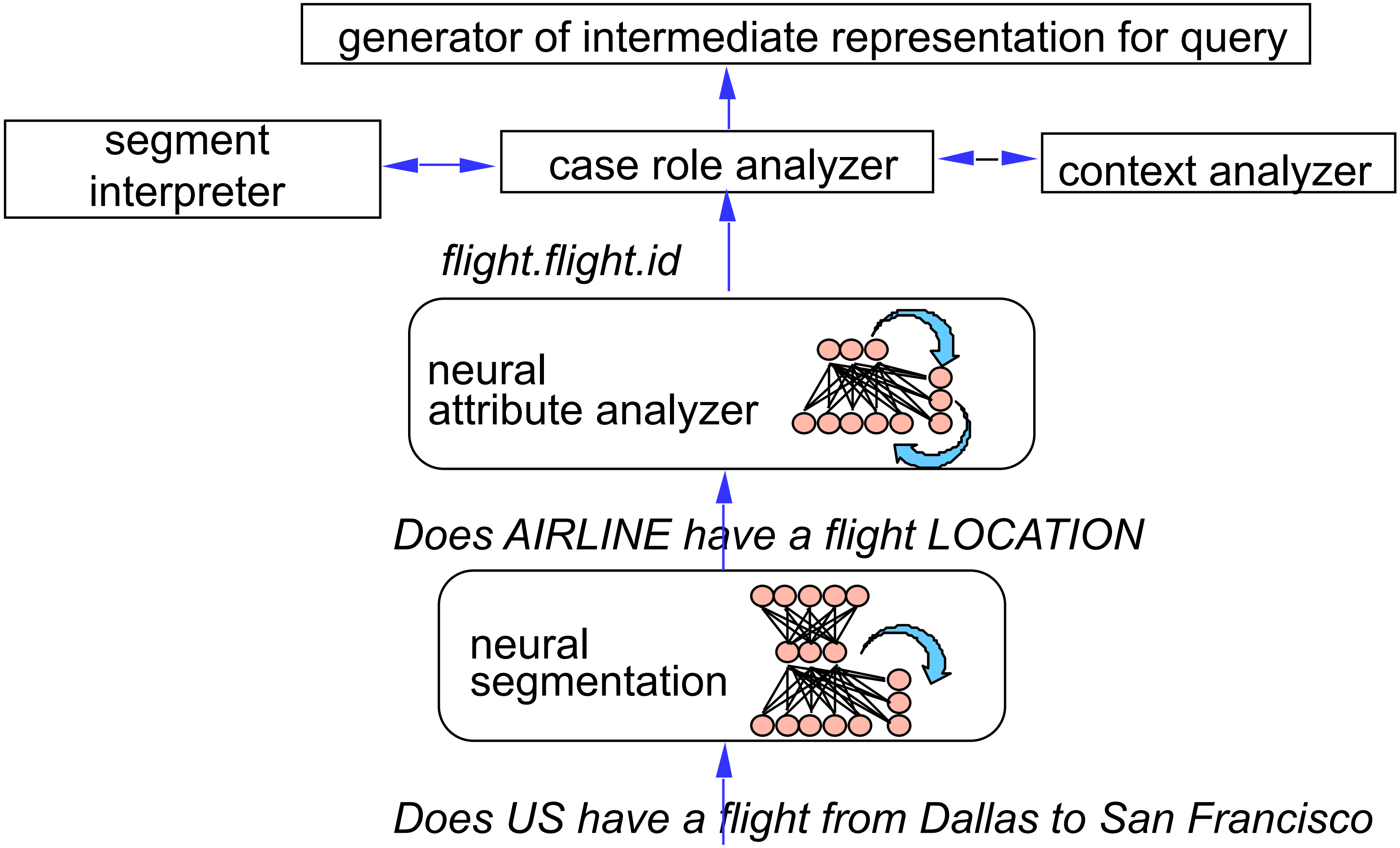
Extracted transducer from a RNN



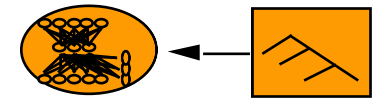
Hybrid processing architectures

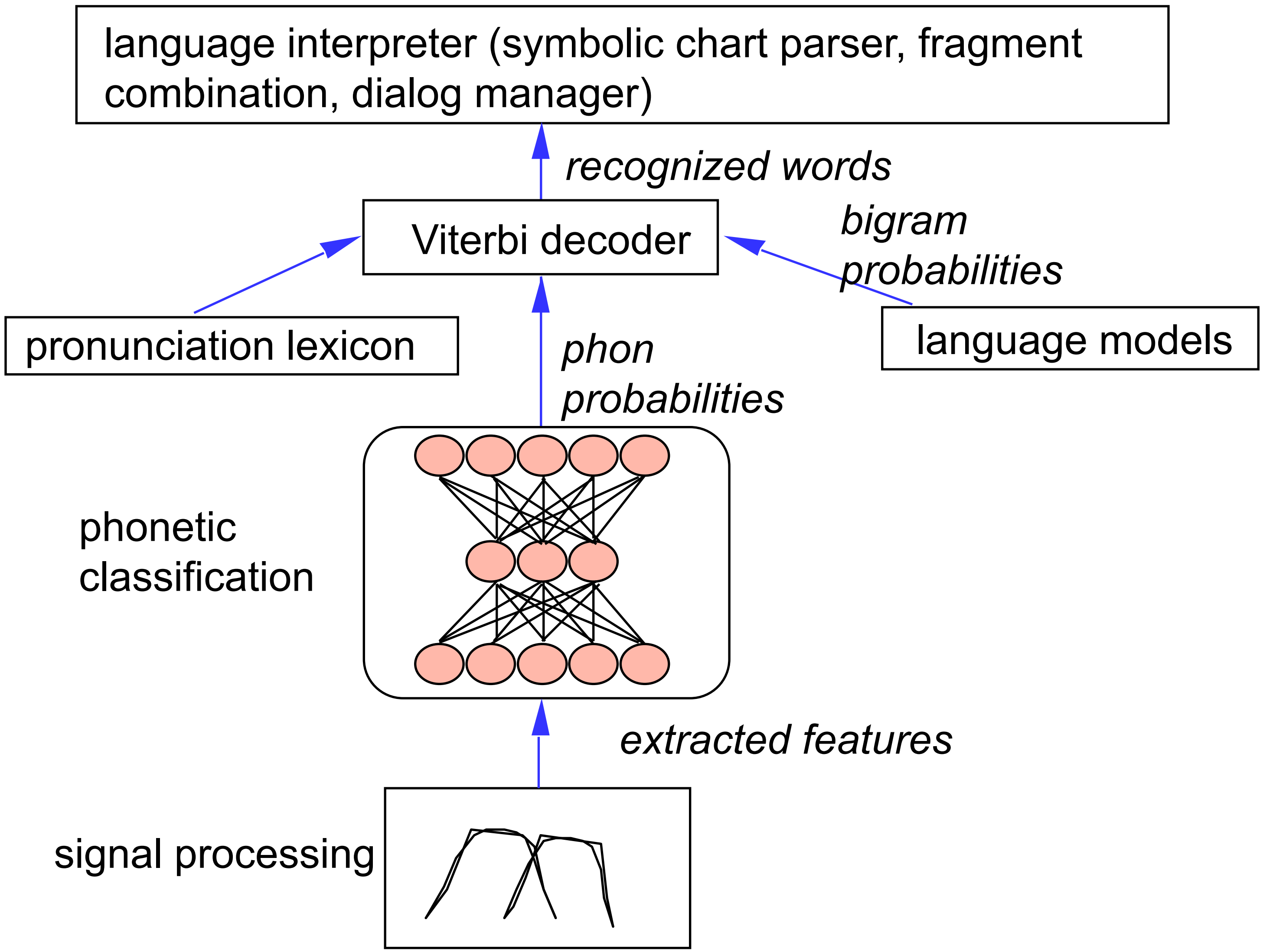
- Consist of symbolic and neural representations at the same time
- Different forms of combination and integration
  - ***loosely coupled***: symbolic and neural modules separate and unidirectional communication
    
  - ***tightly coupled***: symbolic and neural modules separate and bidirectional communication
    
  - ***integrated***: symbolic and neural modules fully embedded and integrated
    

Connectionist attribute extraction and symbolic query construction

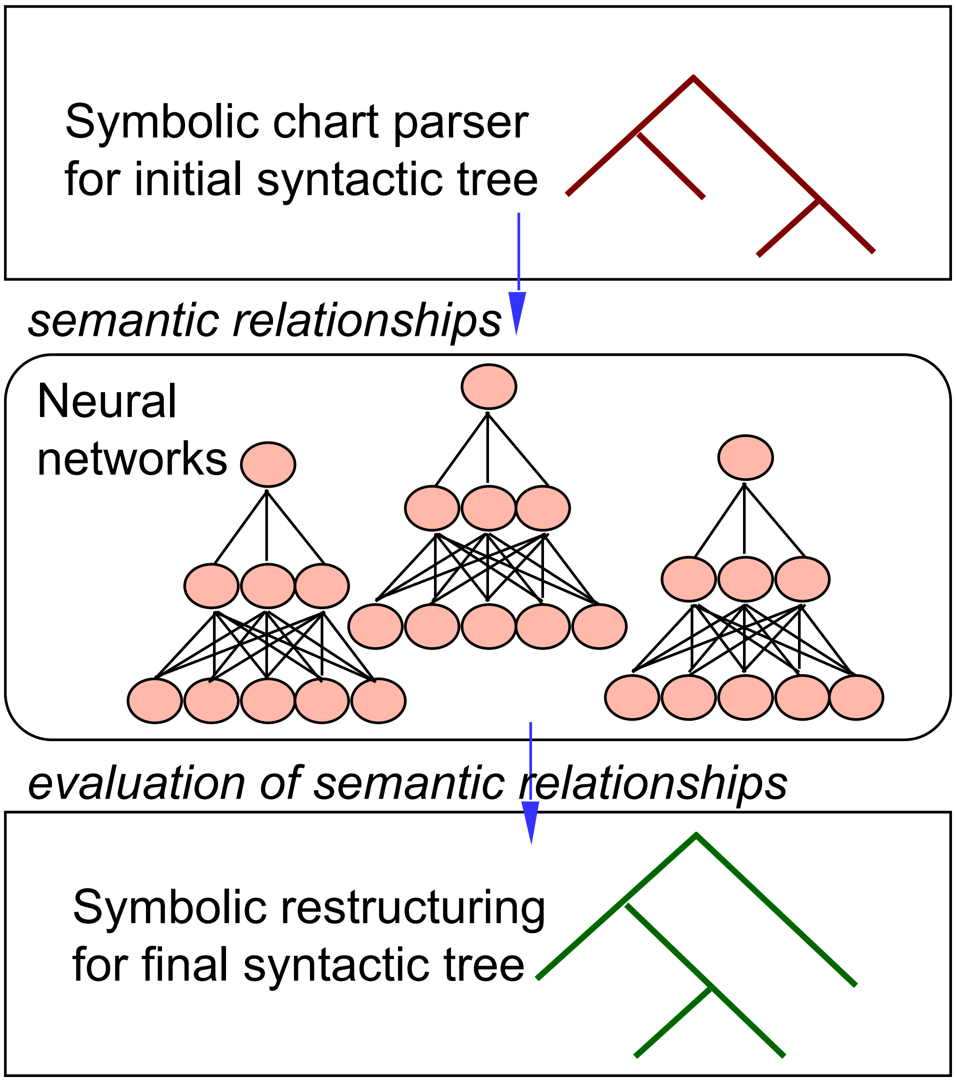


Connectionist phoneme classification and symbolic parsing





Phrase parsing in loosely coupled hybrid architecture SCAN



(The spider from (Argentina in the shoe box))

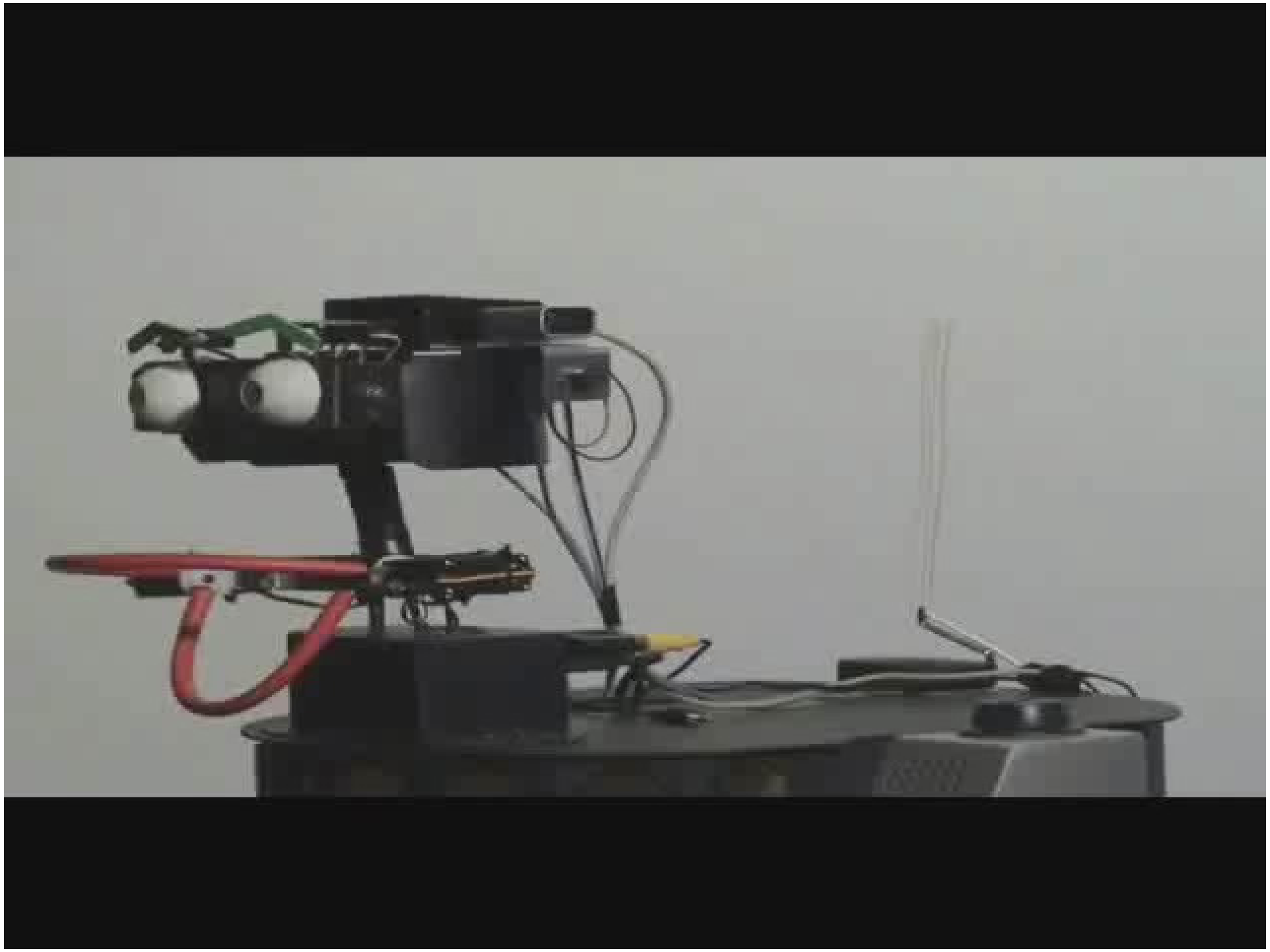
((The spider from Argentina) in the shoe box)

The spider in the shoe box
OK

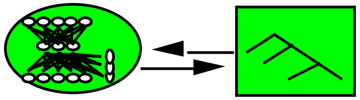
Argentina in the shoe box
NO

((The spider from Argentina) in the shoe box)

Hybrid architecture on MIRA: neural sound localization within symbolic robot architecture



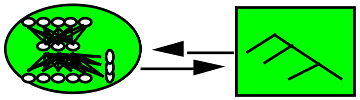
Tightly coupled hybrid processing



- Separate and symbolic and neural modules
- Control flow is ***sequential***
- Communication ***intern*** with common data structures

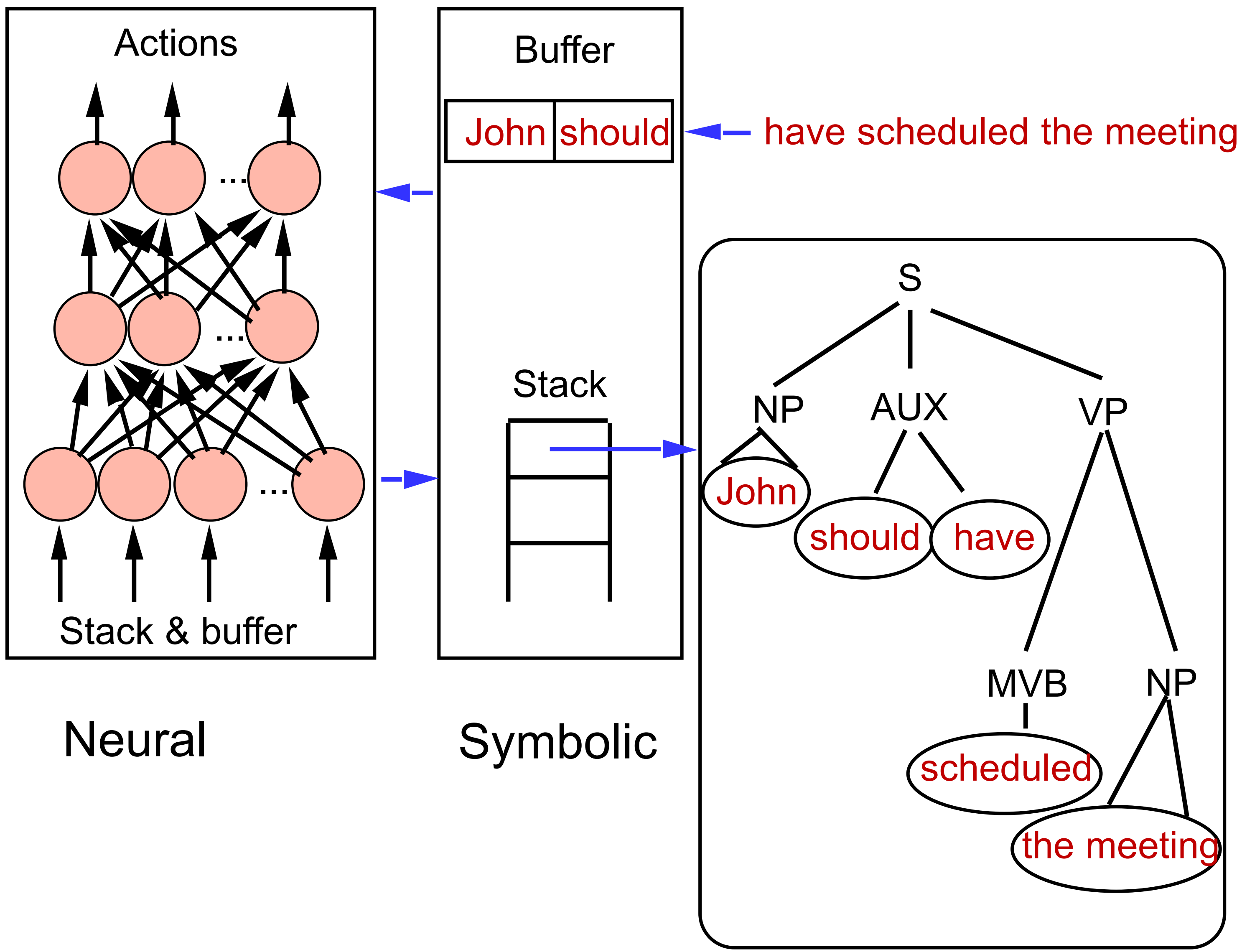
- Main difference to loose coupling:
  - ***bidirectional*** communication
  - influence of symbolic / neural module before neural / symbolic module finishes processing
  - ***interaction*** possible

Tightly coupled deterministic parsing

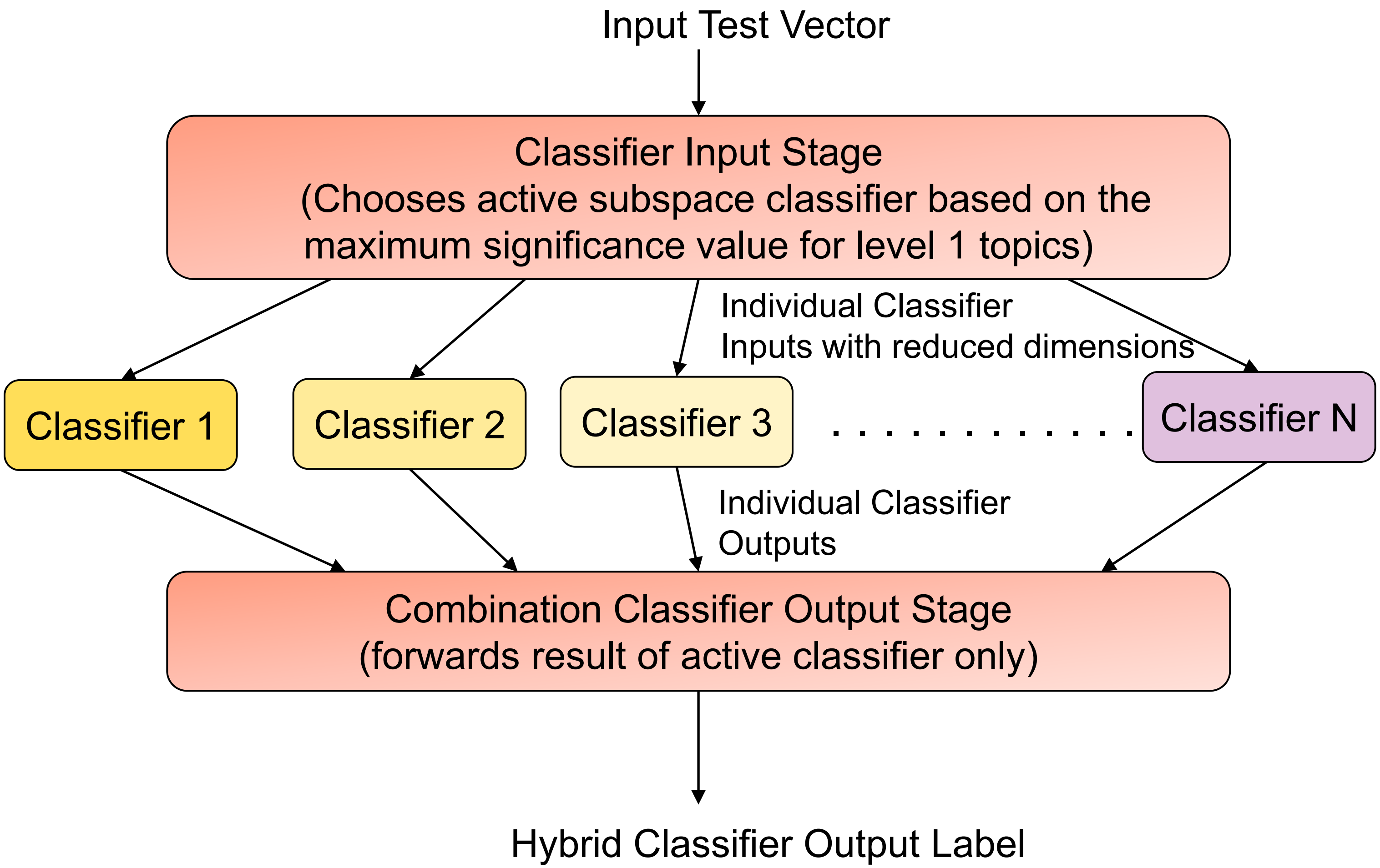


- Coupling of symbolic deterministic parser with neural control module
- Symbolic component processes input sentence
    - puts constituents into buffer or onto stack
- Neural component is feedforward network for finding actions for control
- ***Many interactions*** between symbolic and neural components

Tightly coupled deterministic parsing



Integrated classifier architectures



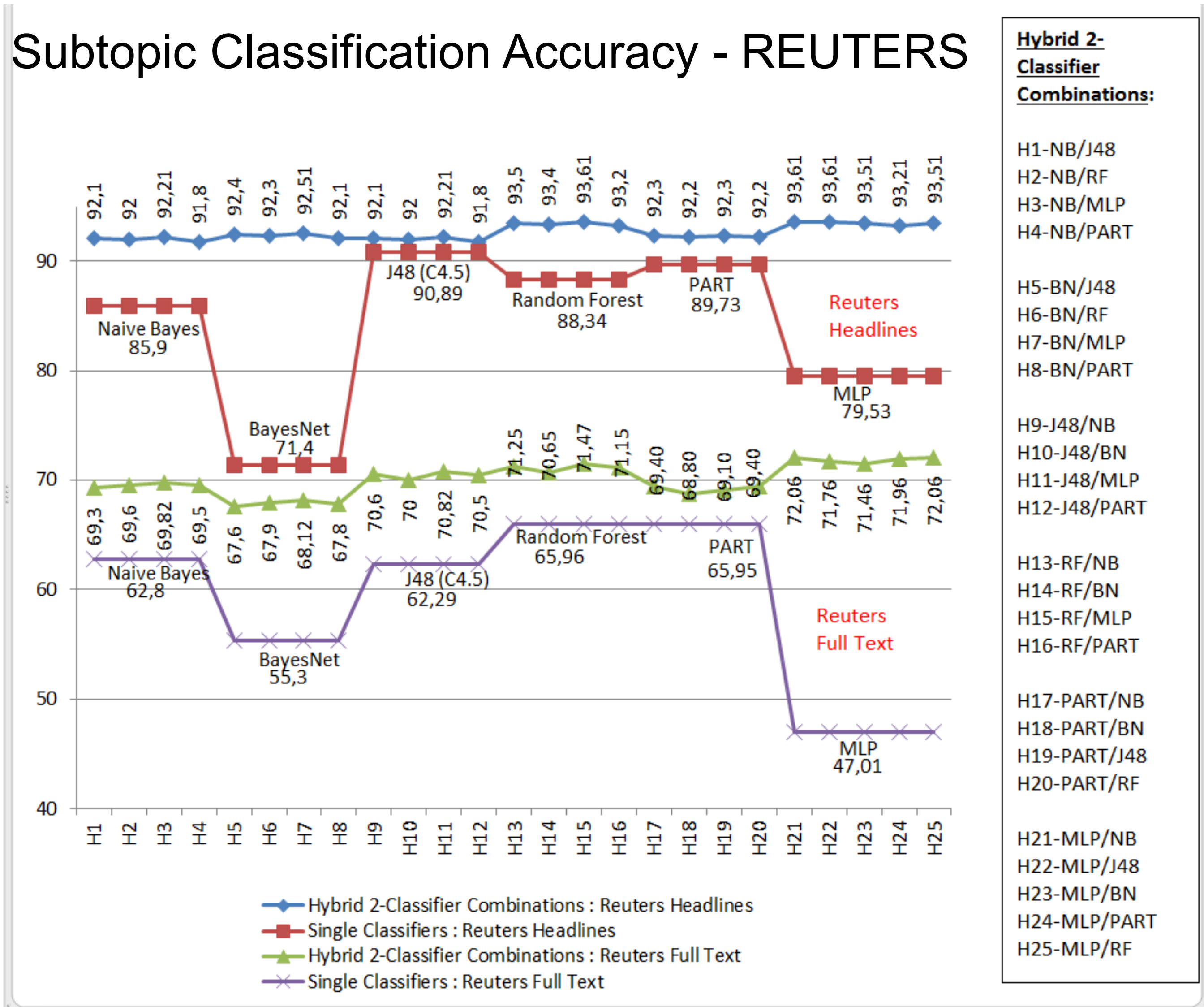
[Tripathi, Oakes, Wermter 2012, 2013]

Test corpora

- Reuters RCV1 News Corpus
  - Reuters Headlines
  - Reuters Full Text (Headlines + Body Text)
  - 4 main topics, 50 subtopics

- LSHTC (Large Scale Hierarchical Text Classification) Corpus
  - From LSHTC challenge associated with the European Conference on Information Retrieval (ECIR), 2010
  - derived from the ODP (Open Directory Project) directory
  - data in the form of content vectors
  - 10 main topics, 158 subtopics

Hybrid two-classifier combinations



Further reading

- Wermter S. The Hybrid Approach to Artificial Neural Network-based Language Processing. In: Dale R., Moisl H. and Somers H. (Ed.) *Handbook of Natural Language Processing*. p. 823-846. Marcel Dekker. 2000.
- Wermter S. Knowledge Extraction from Transducer Neural Networks. *Journal of Applied Intelligence.* Vol. 12, p. 27-42. 2000.
- Tripathi, N., Oakes, M., Wermter, S. Hybrid classifiers based on semantic data subspaces for two-level text categorization. *International Journal of Hybrid Intelligent Systems*,Vol. 10, pp. 33–41, IOS Press Amsterdam, 2013.