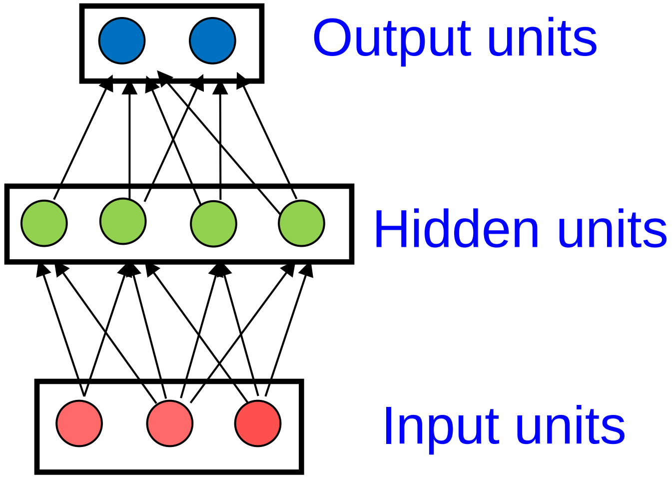
Neural Networks

Lecture 5: Sequence Learning

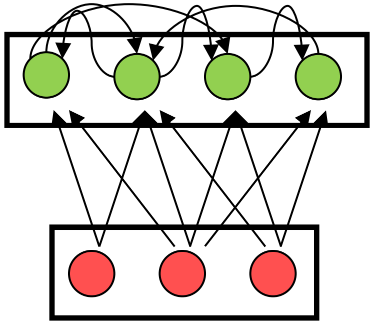
http://www.informatik.uni-hamburg.de/WTM/

Revision:
Types of connectivity

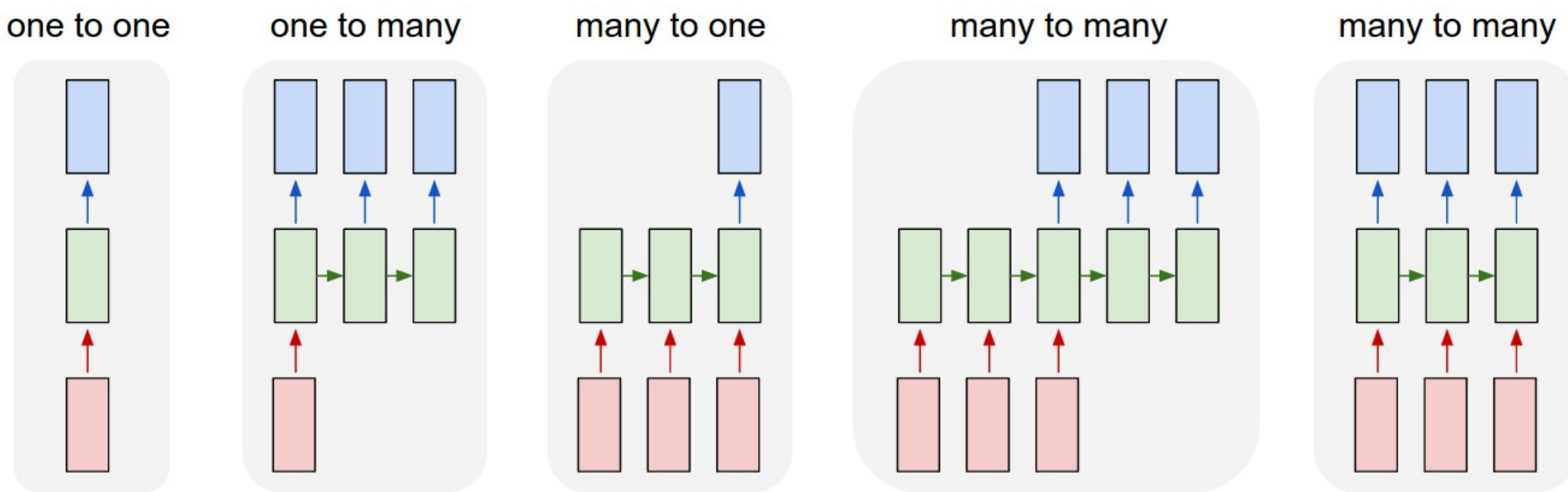
- Feedforward networks
  - Compute a series of transformations
  - Typically, first layer is input and last layer is output
  - Efficient mappings



- Recurrent networks
  - Have directed cycles and can have more complex temporal dynamics
  - More biologically realistic



Sequence learning

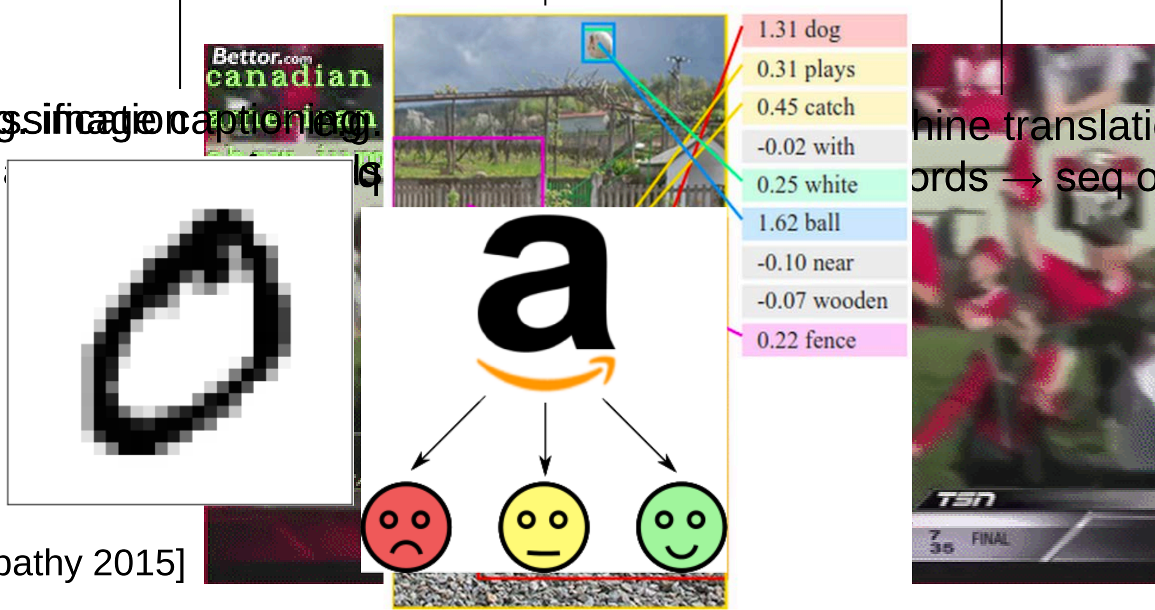


e.g. image classification
(MLP/CNN)

e.g. image captioning

...hine translation
...ords → seq of words

e.g. video classification
on frame level



Based on [Karpathy 2015]

Sequence learning

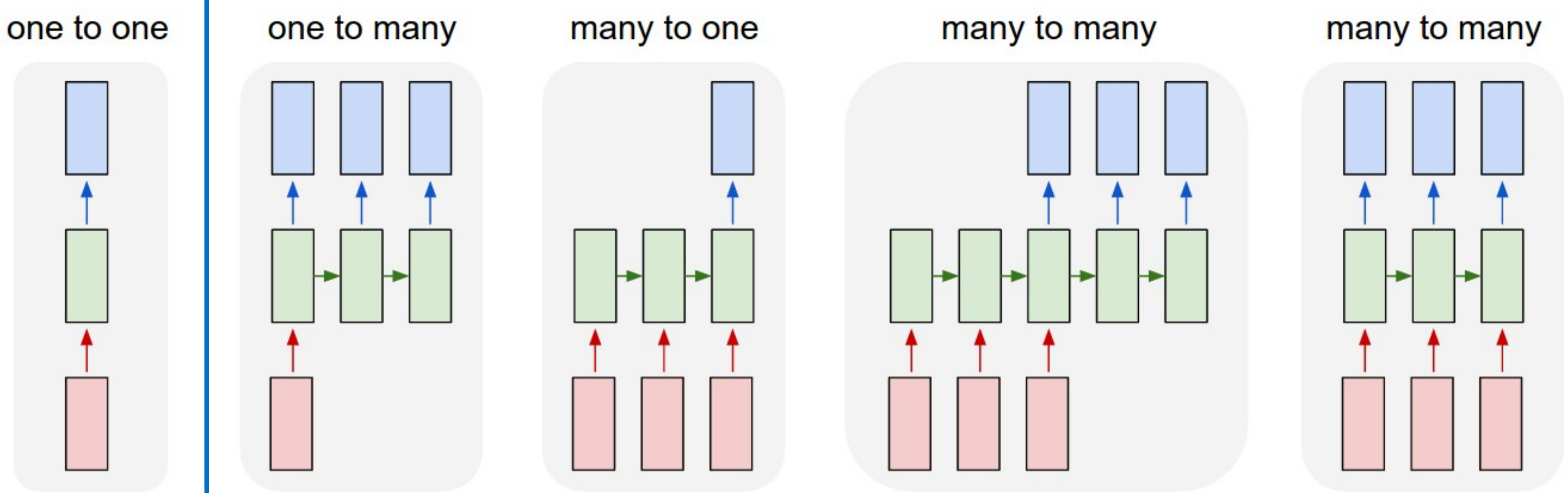


image captioning

machine translation

image classification

sentiment analysis

video classification

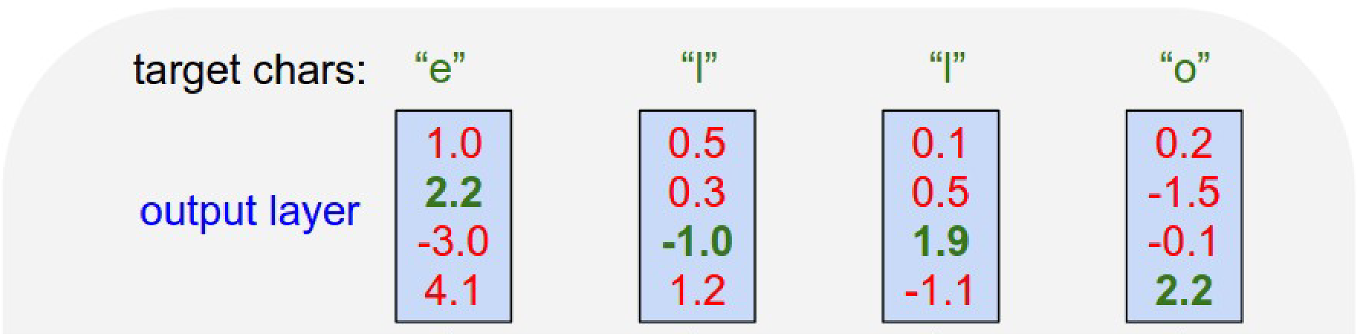
Recurrent Neural Networks

Based on [Karpathy 2015]

Softmax

- Usually desirable in classification to have output vector model the joint probability distribution

- For classification, we may have some generated output values using the cross-entropy loss:



- This can be normalized with softmax so that the values are in [0,1] and add up to 1

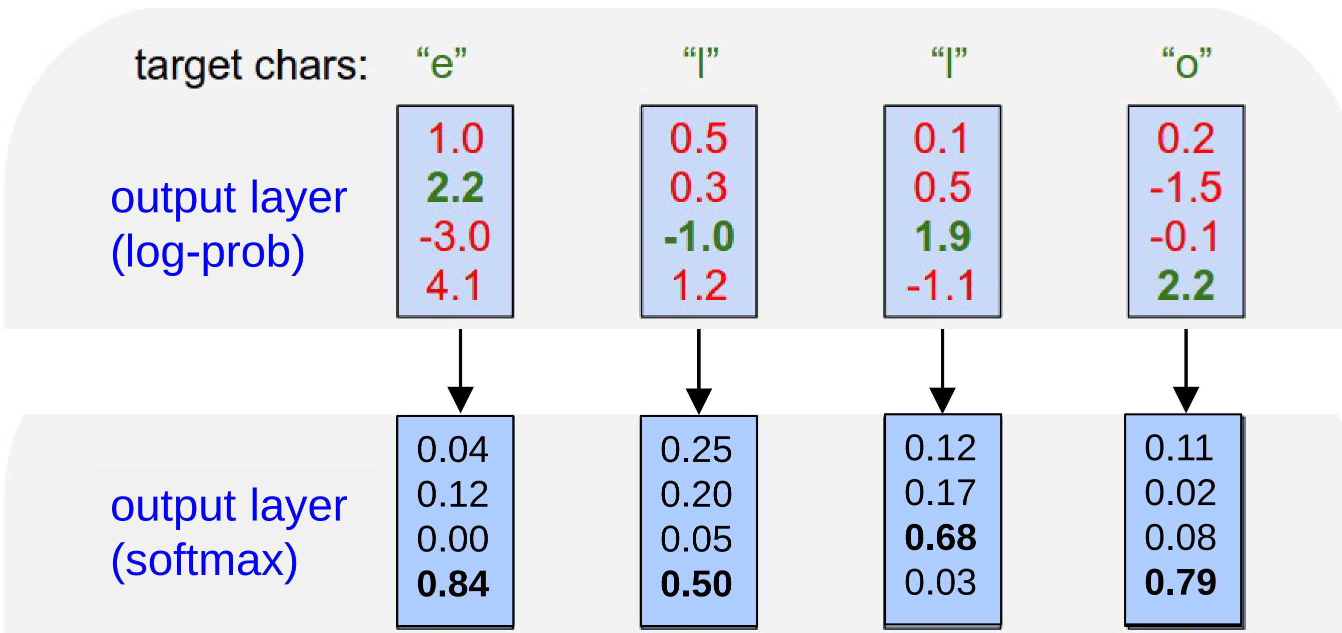
Softmax

$$\sigma(\boldsymbol{z_i}) = \frac{\exp(\boldsymbol{z_i})}{\sum_{j \in I_{all}} \exp(\boldsymbol{z_j})}$$

log-prob. to prob.

normalize

```
y = np.array([1, 2.2, -3, 4.1])
e = np.exp(y)
print  e / np.sum(e)
>>> [0.04, 0.12, 0., 0.84]
```



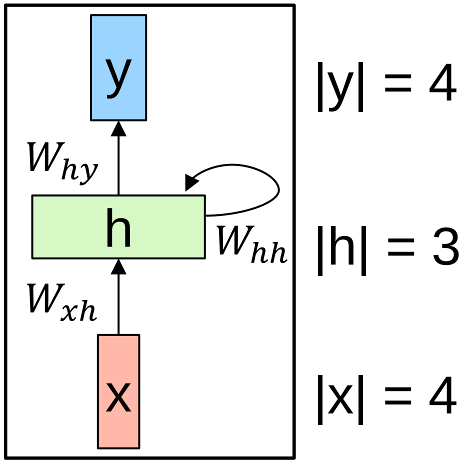
- Drawback: Computationally expensive for very large vectors ($\exp$)

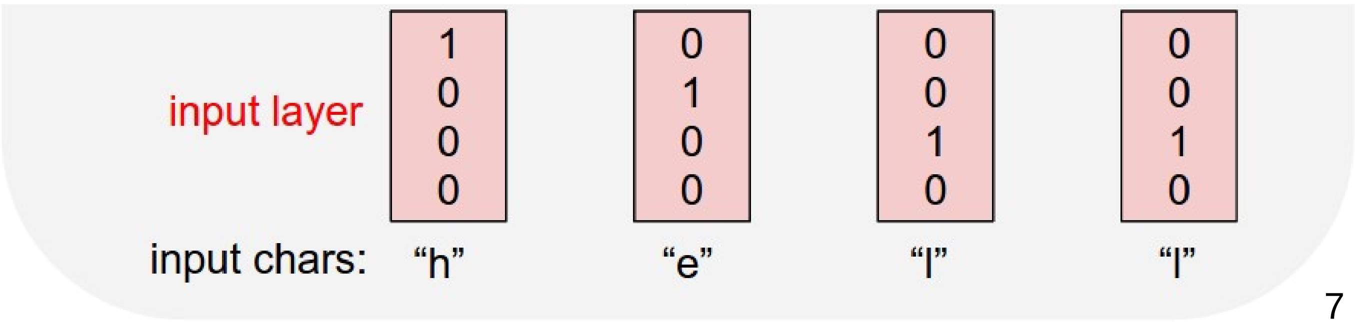
Example:
Character-level language model for prediction

- Vocabulary: [h,e,l,o]

- 1 unit per char: “one-hot” encoding

- Example training sequence: **“hello”**





Example continued: Generating text with character-based recurrent network (char-rnn):

- 100 iterations:

> tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e plia tklrgd t o idoe ns,smtt h ne etie h,hregtrs nigtike,aoaenns lng

- 300 iterations:

> "Tmont thithey" fomesscerliund Keushey. Thom here sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."

- 500 iterations:

> we counter. He stutn co des. His stanted out one ofler that concossions and was to gearang reay Jotrets and with fre colt otf paitt thin wall. Which das stimn

[github.com/karpathy/char-rnn] (Trained on Leo Tolstoy's "War and Peace")

Example continued: Generating text with character-based recurrent network (char-rnn):

- 700 iterations:

  > Aftair fall unsuch that the hall for Prince Velzonski's that me of her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort how, and Gogition is so overelical and ofter.

- 1200 iterations:

  > "Kite vouch!" he repeated by her door. "But I would be done and quarts, feeling, then, son is people...."

- 2000 iterations:

  > "Why do what that day," replied Natasha, and wishing to himself the fact the princess, Princess Mary was easier, fed in had oftened him. Pierre aking his soul came to the packs and drove up his father-in-law women

[github.com/karpathy/char-rnn] (Trained on Leo Tolstoy's "War and Peace")

Why we want word-level learning rather than character-level

- Word-level:

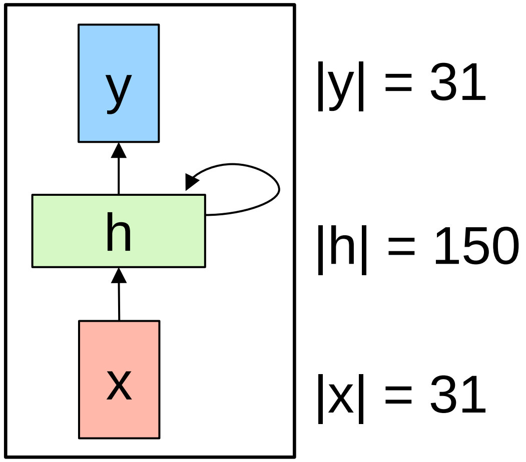
“run”, “runs”, “running”:
lexeme=“run”, morpheme=“-s”, “-ing”

  + Predefined structure
  + Most NLP algorithms operate on words
  – Word segmentation into lexemes and morphemes difficult
  – Word boundary detection

- Char-level:
  + Learning word segmentation as a side effect
  + char2word mapping possible as intermediate step
  – No predefined structure, very long sequences

How can an SRN learn structure of lexical classes from word order?

- Humans learn from constrained ordered words
- Can a network learn structure from order?
- Sentence generator based on categories of lexical items
- Example: Each word represented by different 31 bit vector: one bit is on if word present (one-hot/localist encoding)
- 27,354 word vectors in the 10,000 sentences were concatenated



Categories of lexical items

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

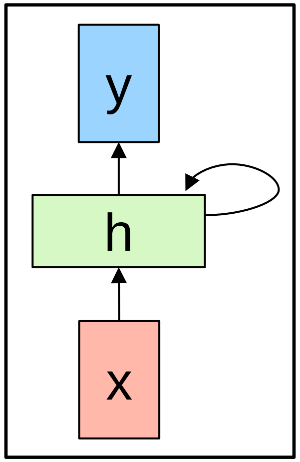
Templates for sentence generator

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

Learning to predict successive words

INPUT		OUTPUT	
0000000000000000000000000000010	(woman)	0000000000000000000000000010000	(smash)
0000000000000000000000000010000	(smash)	0000000000000000000001000000000	(plate)
0000000000000000000001000000000	(plate)	0000010000000000000000000000000	(cat)
0000010000000000000000000000000	(cat)	0000000000000000000100000000000	(move)
0000000000000000000100000000000	(move)	0000000000000000100000000000000	(man)
0000000000000000100000000000000	(man)	0001000000000000000000000000000	(break)
0001000000000000000000000000000	(break)	0000100000000000000000000000000	(car)
0000100000000000000000000000000	(car)	0100000000000000000000000000000	(boy)
0100000000000000000000000000000	(boy)	0000000000000000000100000000000	(move)
0000000000000000000100000000000	(move)	0000000000001000000000000000000	(girl)
0000000000001000000000000000000	(girl)	0000000000100000000000000000000	(eat)
0000000000100000000000000000000	(eat)	0010000000000000000000000000000	(bread)
0010000000000000000000000000000	(bread)	0000000010000000000000000000000	(dog)
0000000010000000000000000000000	(dog)	0000000000000000000100000000000	(move)
0000000000000000000100000000000	(move)	0000000000000000001000000000000	(mouse)
0000000000000000001000000000000	(mouse)	0000000000000000001000000000000	(mouse)
0000000000000000001000000000000	(mouse)	0000000000000000000100000000000	(move)
0000000000000000000100000000000	(move)	1000000000000000000000000000000	(book)
1000000000000000000000000000000	(book)	0000000000000001000000000000000	(lion

Hierarchical cluster analysis of hidden layers: -- similarity structure emerges

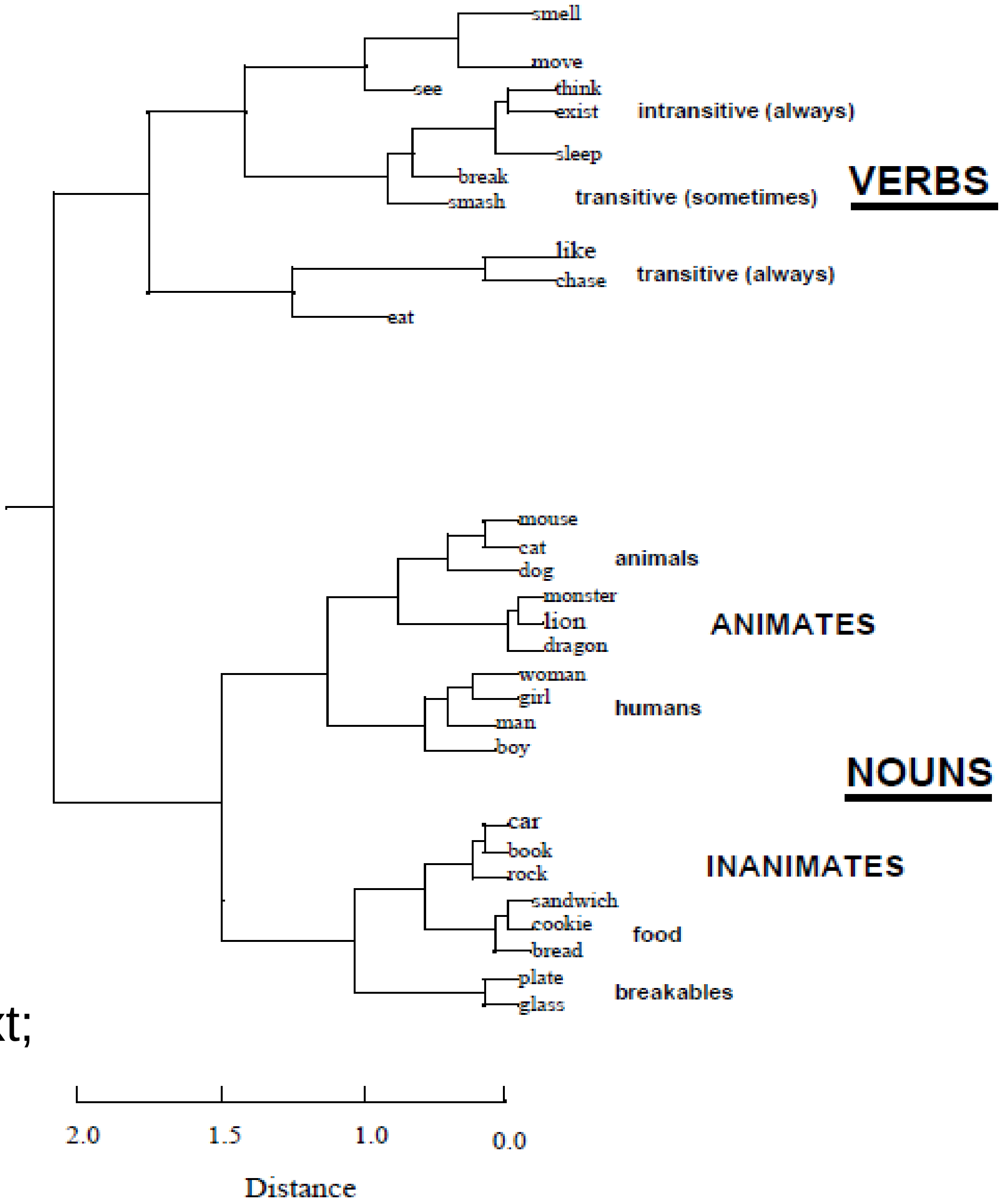


|y| = 31

|h| = 150

|x| = 31

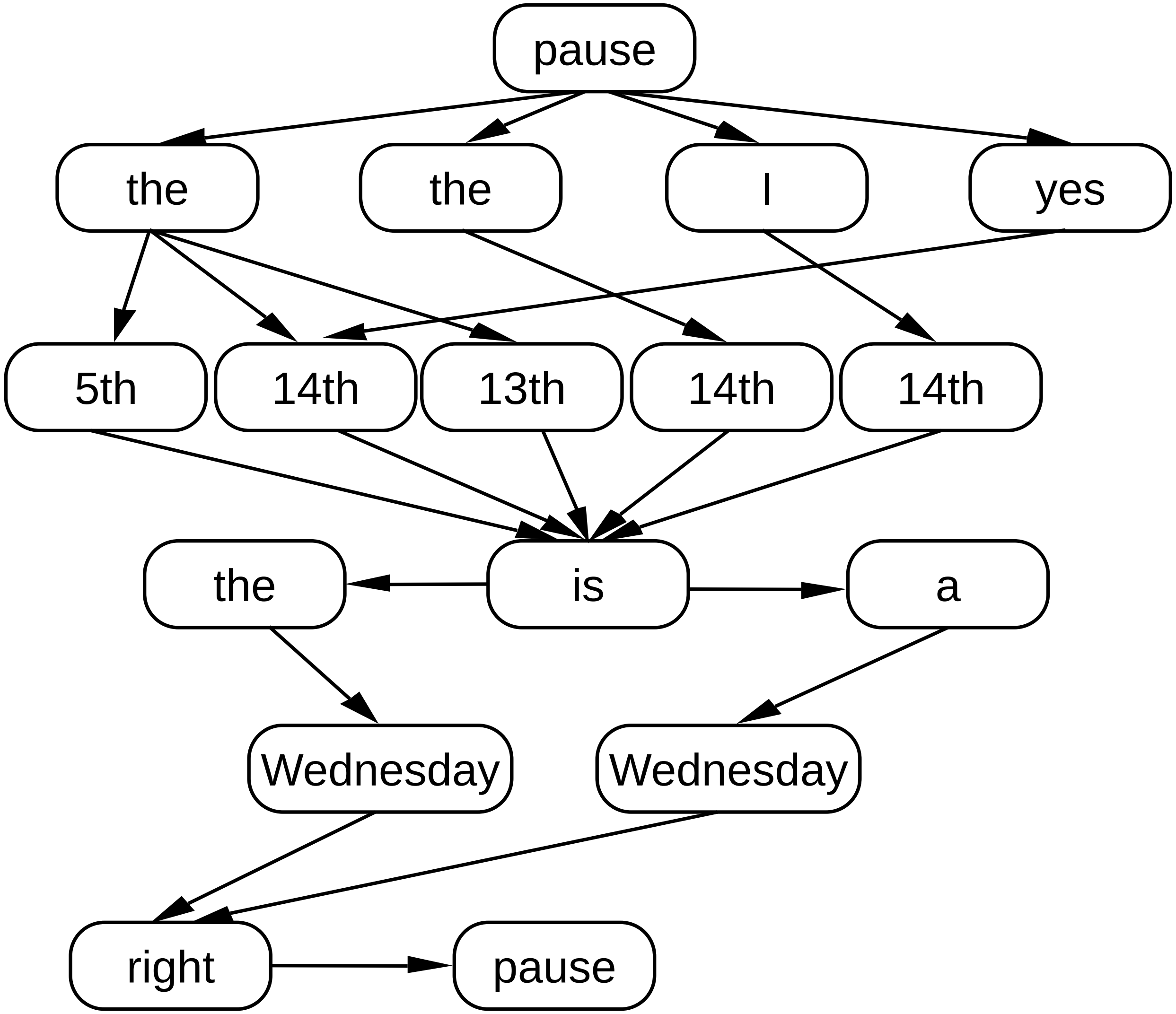
inputs were presented in context; hidden unit vectors averaged across multiple contexts.



How can SRN deal with “noisy” sequences?

- Spoken language is very noisy
- “Incorrect” grammatical constructions
- Interjections and pauses (eh, ah)
- Word repairs, phrase repairs (in - on the table)
- False starts
- Example: I would like eh to call a meeting - pause - a meeting on Wednesday no Thursday at four
- Speech recognizers add even more noise

Simplified word graph from speech recognizer



Dealing with incorrect sequences in recurrent networks

- Basic syntactic categories
  - noun (N), verb (V), preposition (R), pronoun (U), numeral (M), participle (P), pause (/), adjective (J), adverb (A), conjunction (C), determiner (D), interjection (I), other (O)
- Abstract syntactic categories
  - noun group (NG), verb group (VG), adverbial group (AG), prepositional group (PG), conjunction group (CG), modus group (MG), special group (SG), interjection group (IG)
- Goal: Learning a robust flat analysis at the level of phrasal syntactic categories
- **Example**:

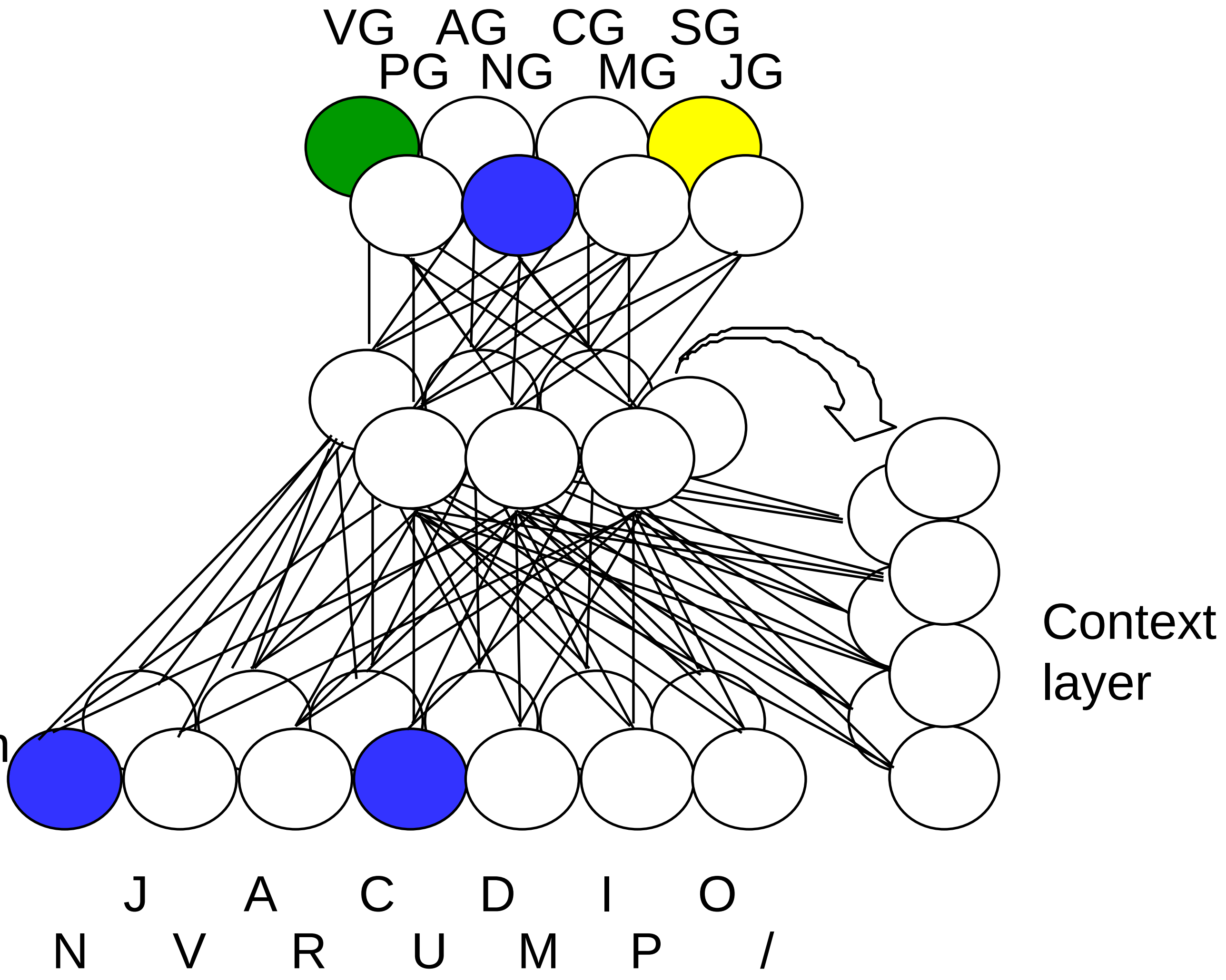
I	would	suggest	eh	a	meeting	on	Friday
U	V	V	I	D	VN	R	N
NG	VG		IG	NG		PG	

Dealing with incorrect sequences in recurrent networks

Output layer

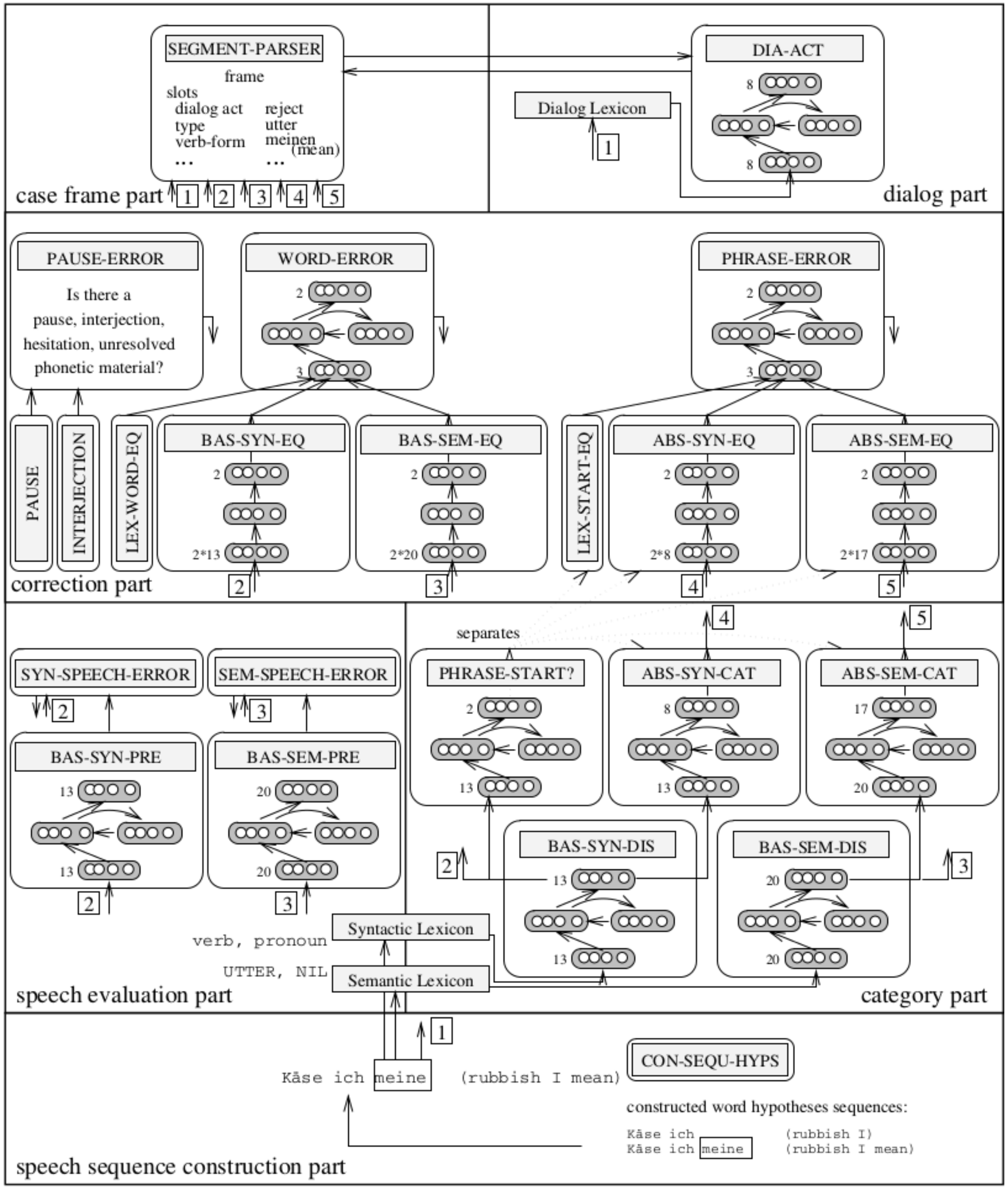
Hidden layer

Input layer: static representation for word **w**



Interpretation of system performance based on activation of networks

- Building larger architectures based on simple recurrent networks
- Detailed interpretation of learning capabilities
- Detailed interpretation of performance
- Example systems: SCREEN (Wermter, Weber 1997)
  (Symbolic Connectionist Robust EnterprisE for Natural language)



SCREEN

architecture details

https://www2.informatik.uni-hamburg.de/wtm/ps/jair97.pdf

Examples for basic and abstract syntactic categories

Category	Examples
noun (N)	date, April
adjective (J)	late
verb (V)	meet, choose
adverb (A)	often
preposition (R)	at, in
conjunction (C)	and, but
pronoun (U)	I, you
determiner (D)	the, a
numeral (M)	fourteenth
interjection (I)	eh, oh
participle (P)	taken
other (O)	particles
pause (/)	pause

Category	Examples
verb group (VG)	mean, would propose
noun group (NG)	a date, the next possible slot
adverbial group (AG)	later, as early as possible
prepositional group (PG)	in the dining hall
conjunction group (CG)	and, either ... or
modus group (MG)	interrogatives, confirmations: when, how long, yes
special group (SG)	additives like politeness: please, then
interjection group (IG)	interjections, pauses: eh, oh

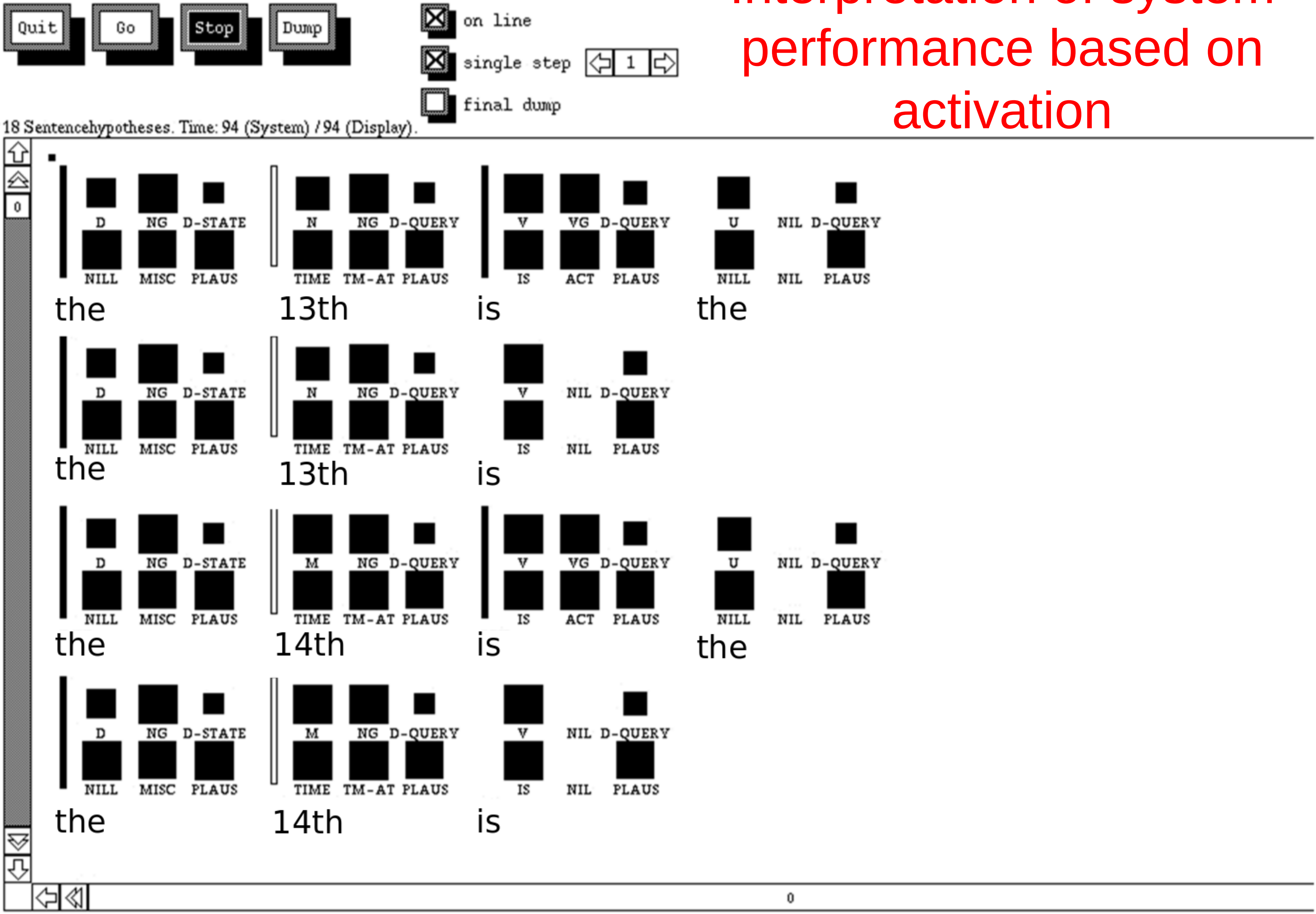
Basic semantic …

Category	Examples
select (SEL)	select, choose
suggest (SUG)	propose, suggest
meet (MEET)	meet, join
utter (UTTER)	say, think
is (IS)	is, was
have (HAVE)	had, have
move (MOVE)	come, go
aux (AUX)	would, could
question (QUEST)	question words: where, when
physical (PHYS)	physical objects: building, office
animate (ANIM)	animate objects: I, you
abstract (ABS)	abstract objects: date
here (HERE)	time or location state words, prepositions: at, in
source (SRC)	time or location source words, prepositions: from
destination (DEST)	time or location destination words, prepositions: to
location (LOC)	Hamburg, Pittsburgh
time (TIME)	tomorrow, at 3 o'clock, April
negative evaluation (NO)	no, bad
positive evaluation (YES)	yes, good
nil (NIL)	words "without" specific semantics, e.g. determiner: a

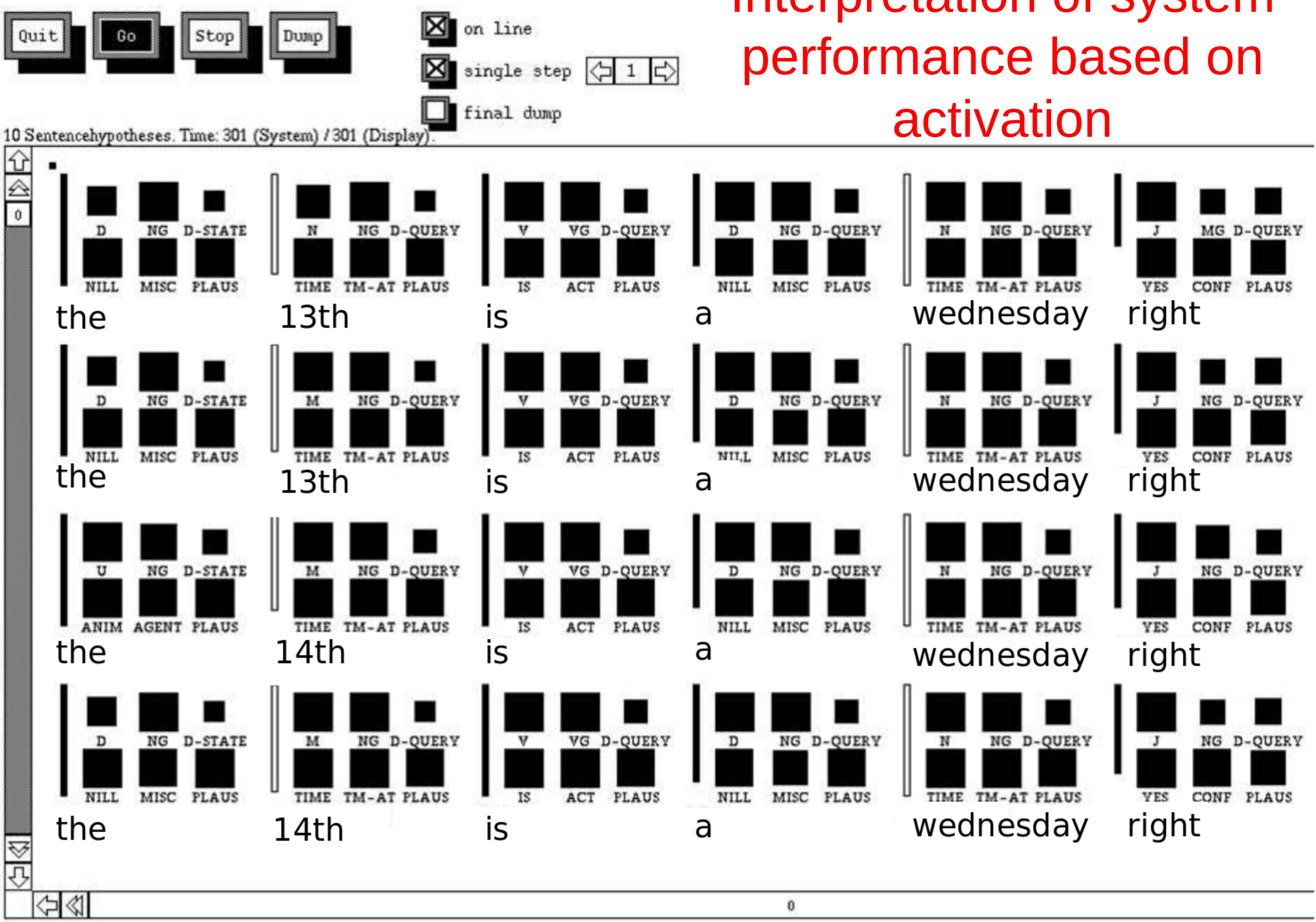
… and abstract semantic categories

Category	Examples
action (ACT)	action for full verb events: meet, select
aux-action (AUX)	auxiliary action for auxiliary events: would like
agent (AGENT)	agent of an action: I
object (OBJ)	object of an action: a date
recipient (RECIP)	recipient of an action: to me
instrument (INSTR)	instrument for an action: using an elevator
manner (MANNER)	how to achieve an action: without changing rooms
time-at (TM-AT)	at what time: in the morning
time-from (TM-FRM)	start time: after 6 am
time-to (TM-TO)	end time: before 8 pm
loc-at (LC-AT)	at which location: in Frankfurt, in New York
loc-from (LC-FRM)	start location: from Boston, from Dortmund
loc-to (LC-TO)	end location: to Hamburg
confirmation (CONF)	confirmation phrase: ok great, yes wonderful
negation (NEG)	negation phrase: no stop, not
question (QUEST)	question phrase: at what time
misc (MISC)	miscellaneous words, e.g., for politeness: please, eh

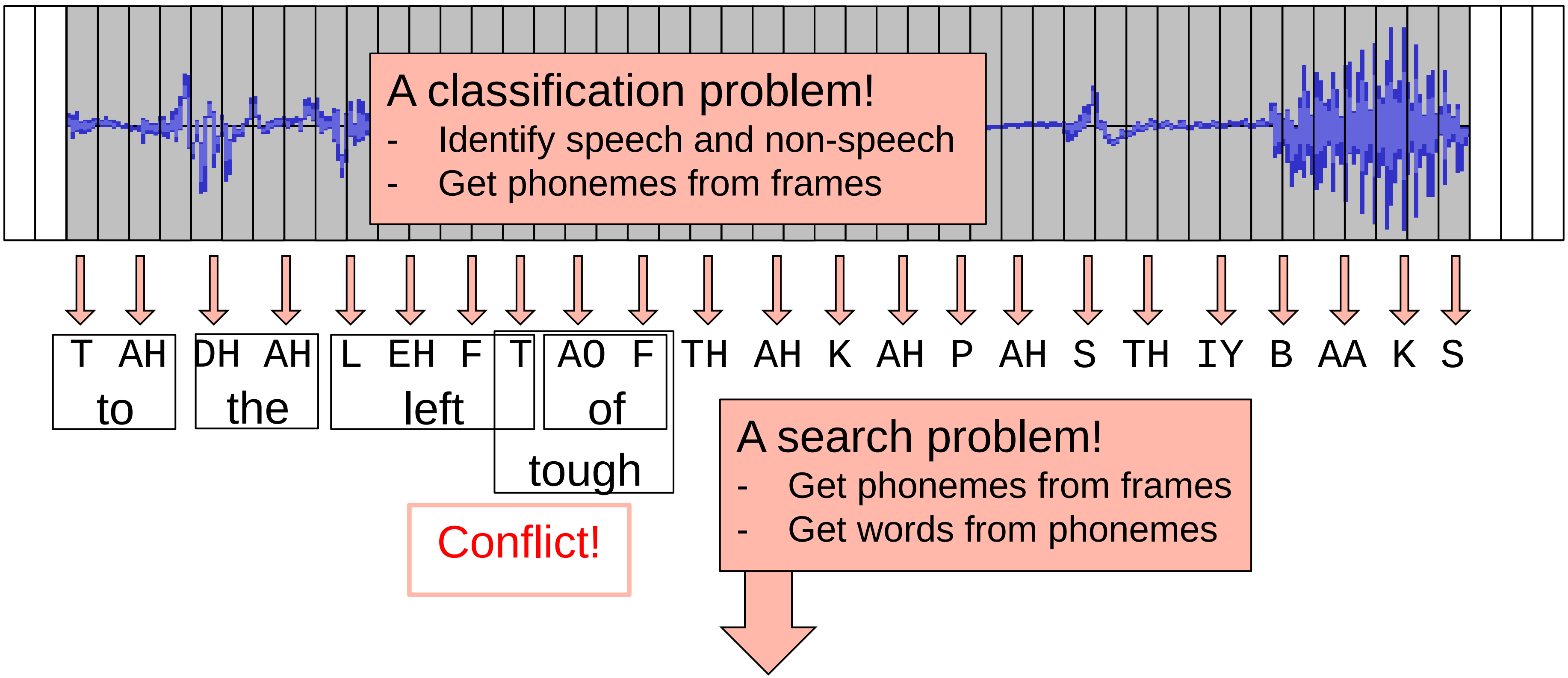
Interpretation of system performance based on activation



Interpretation of system performance based on activation



How to get from sound to the word level at all?

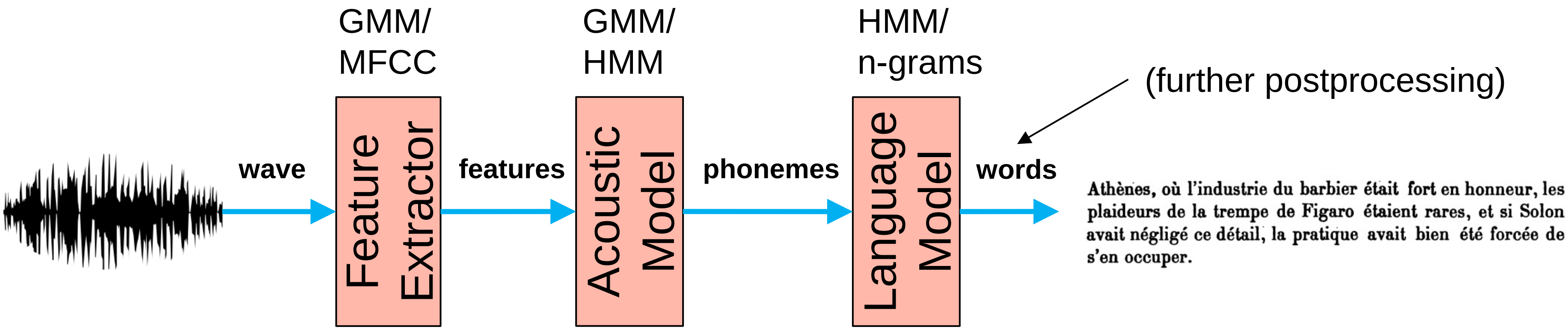


to the left of the cup is the box

Speech Recognition with RNNs

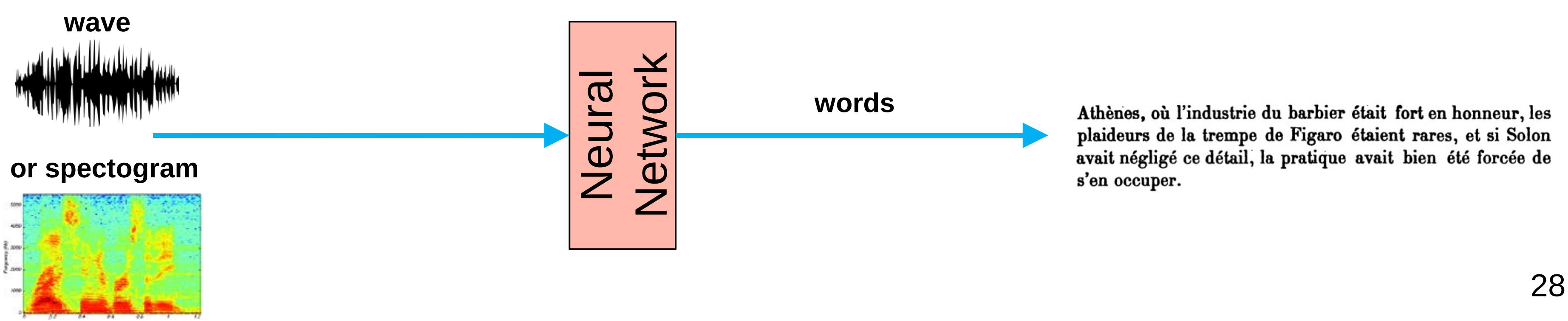
- Traditional Speech Recognition Systems (before ~ 2013):

Sophisticated pipeline of algorithms to get from raw audio to coherent text.



[GMM: Gaussian Mixture Model, HMM: Hidden Markov Model]

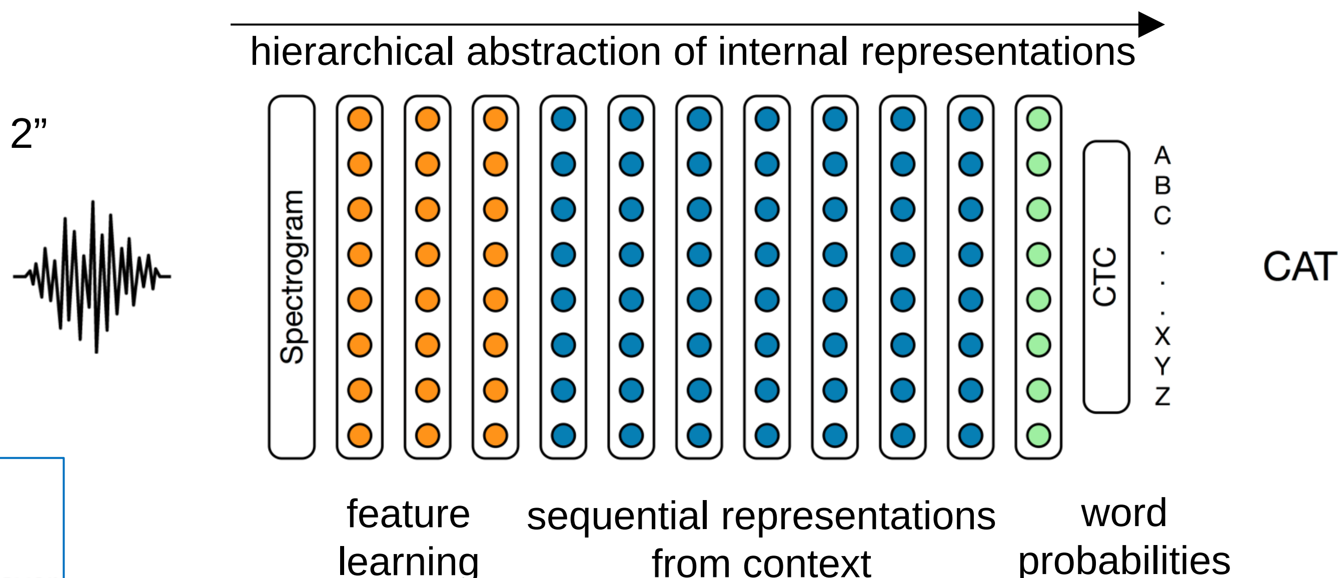
- "End-to-End" Speech Recognition (2013+):

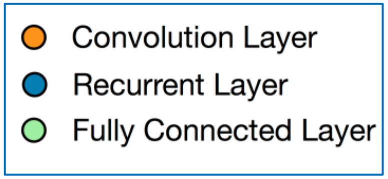


End-to-End Speech Recognition with RNNs

- Neural Networks can:
  - learn features (feature extractor)
  - predict phonemes from features (acoustic model)
  - predict words from phonemes (language model)
- We can implement a network specialized on each task of the pipeline
- Stacking them into a single network is the hard part
- The abstractions we get between the output text and the input sound are now part of a learning box: No phonemes, no hard-coded features

Example:
"Deep Speech 2"
Architecture:

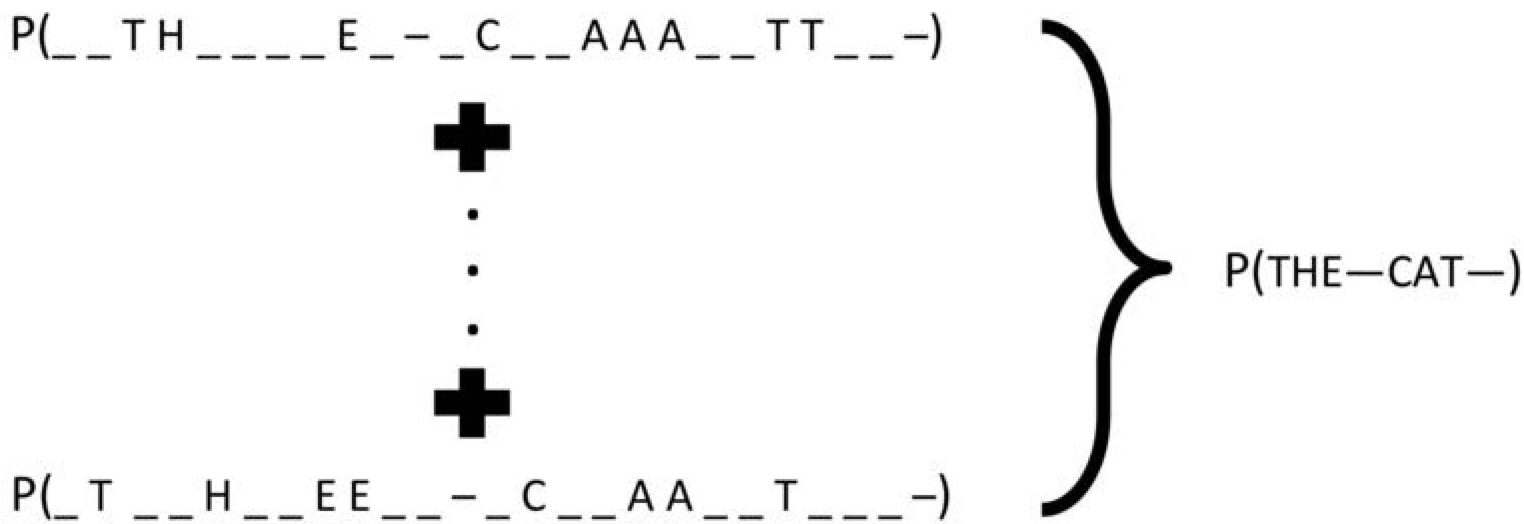




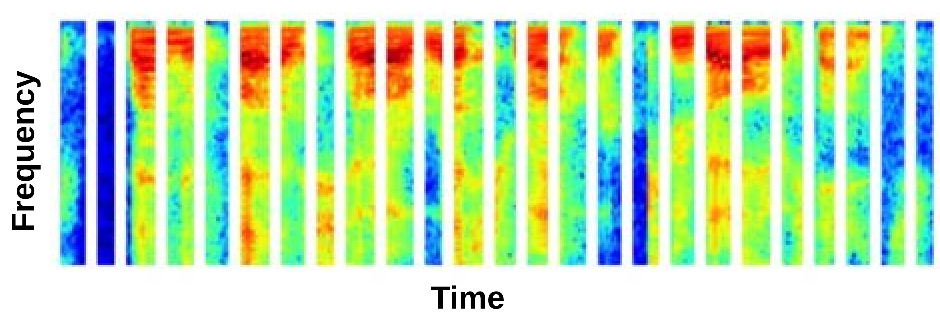
Connectionist Temporal Classification (CTC)

(Graves 2006)

- CTC is a loss function for temporal classification (speech recognition)
- When alignments between inputs and target labels unknown
- Does not require pre-segmented training data



Spectrogram:



Word embeddings

- Millions of words: Need distributed representations!

  hotel (0 0 0 0 0 0 0 0 0 …… 0 0 1 0 0 0 0 0 0 0 0) AND
  motel (0 0 0 0 0 0 0 0 0 …… 0 0 0 0 0 0 1 0 0 0 0) = 0

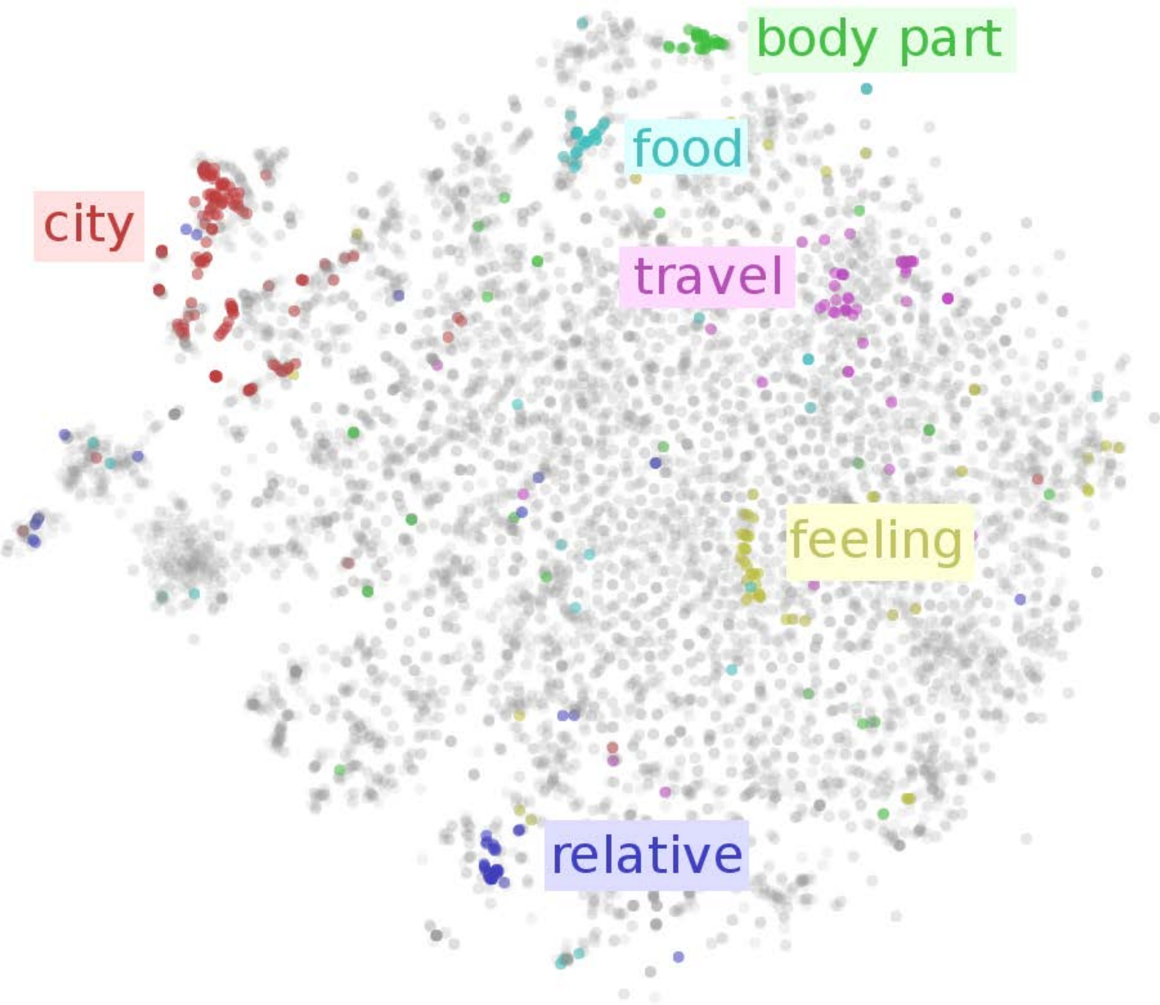
- Approach: Learning word embeddings:
  - Map words to continuous, lower dimensional vectors
  - Captures word meaning in the semantic space

- Resulting word vector should contain linguistic context information, relating it to other words:

vec(Berlin) ≈ vec(Germany) + vec(capital)

vec(queen) ≈ vec(king) – vec(man) + vec(woman)

Lower-dimensional semantic space

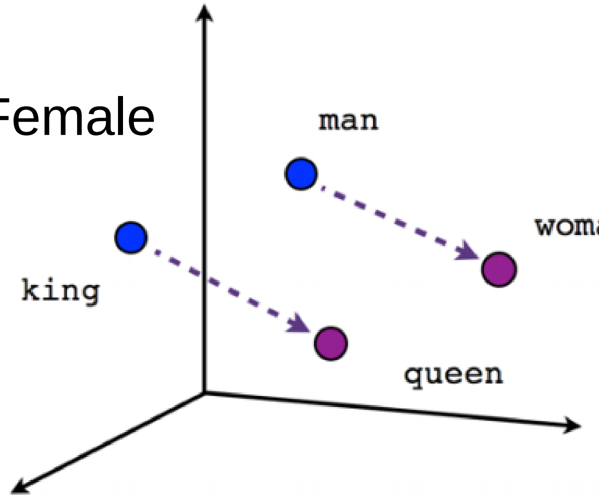


visualized with t-distributed stochastic neighbor embedding (**t-SNE** – van der Maaten and Hinton08)

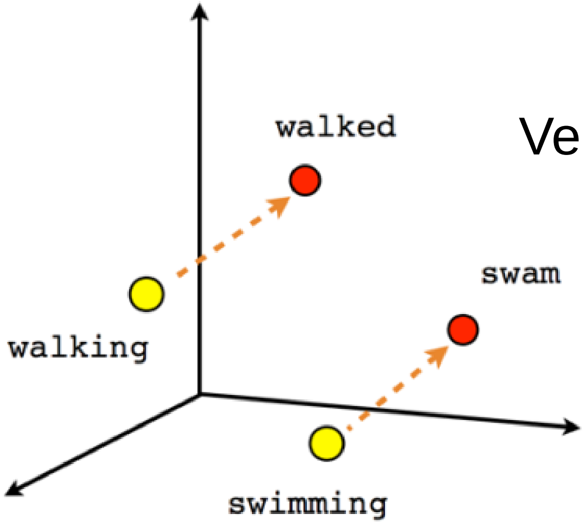
t-SNE: lvdmaaten.github.io/tsne

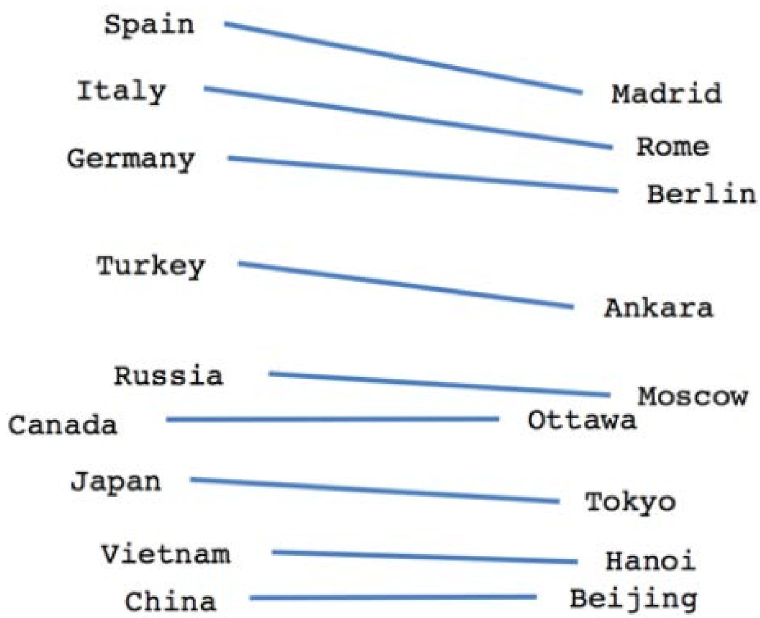
Relationships within the semantic space

Male-Female



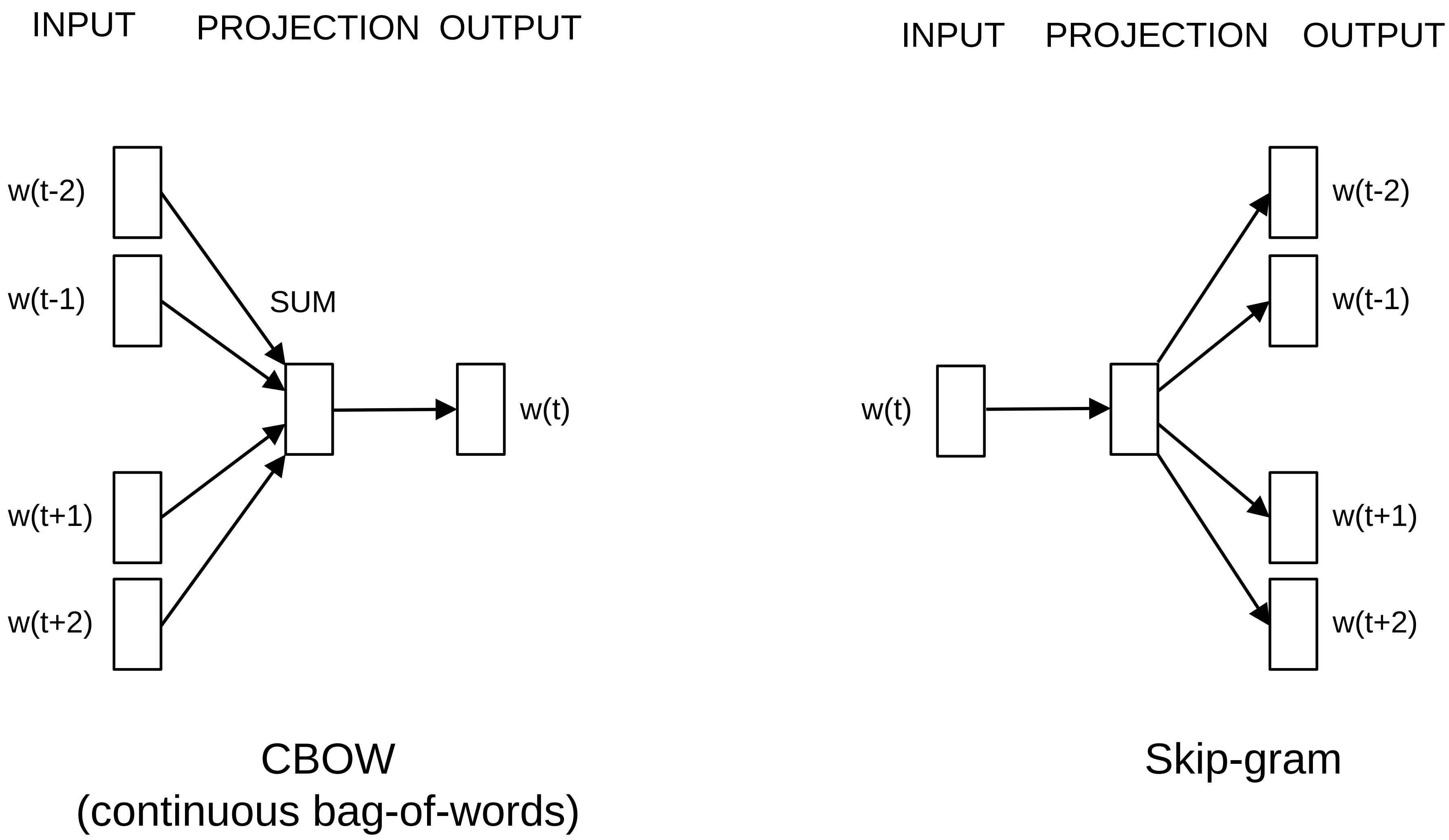
Verb tense





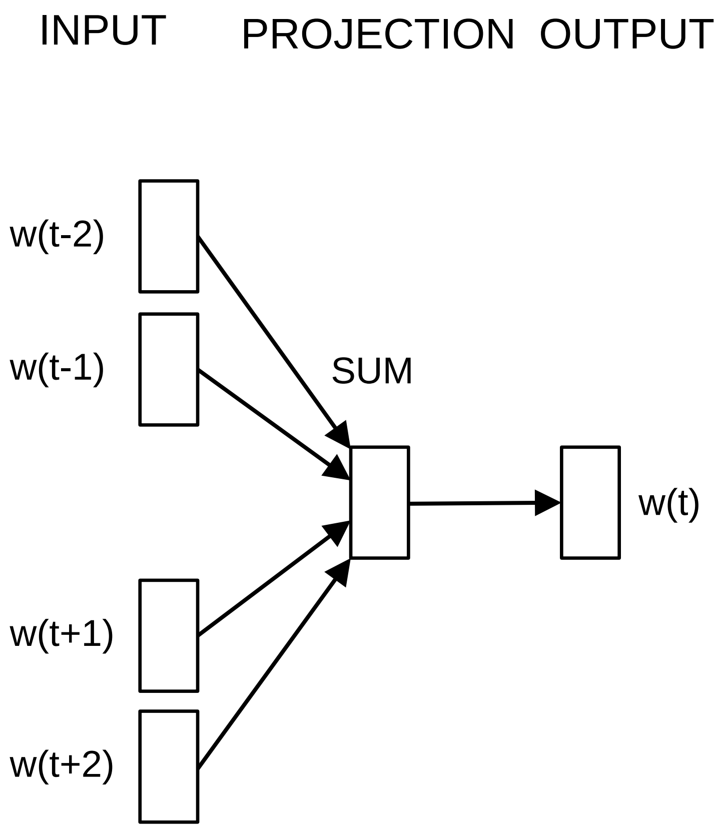
Country-Capital

Learning word embeddings with word2vec



[Mikolov 2013]

Learning word embeddings with word2vec



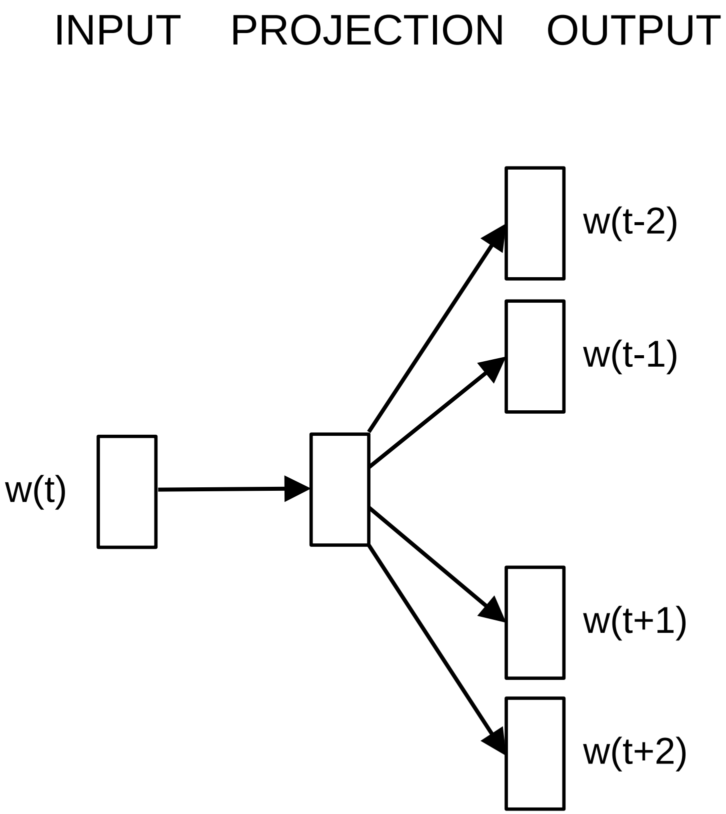
CBOW
(continuous bag-of-words)

- "Given this set of context words, what missing word is also likely to appear?"
- Bag-of-words: order of words does not influence projection
- Projection layer is shared – all words projected on same position (vectors averaged)

[Mikolov 2013]

Learning word embeddings with word2vec

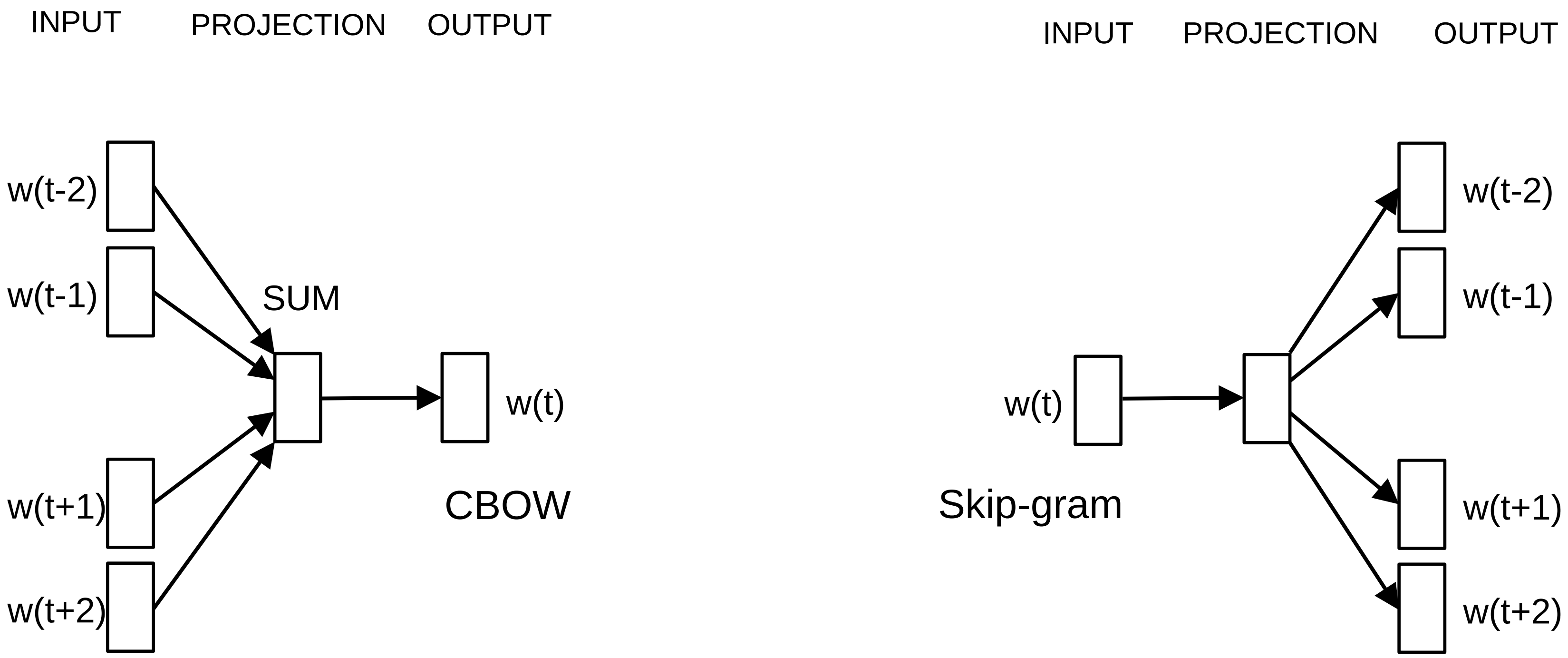
- "Given this single word, what other words are likely to appear with it?"
- Input to classifier with continuous projection layer
- Unlike CBOW, nearby context has more influence than more distant words



Skip-gram

[Mikolov 2013]

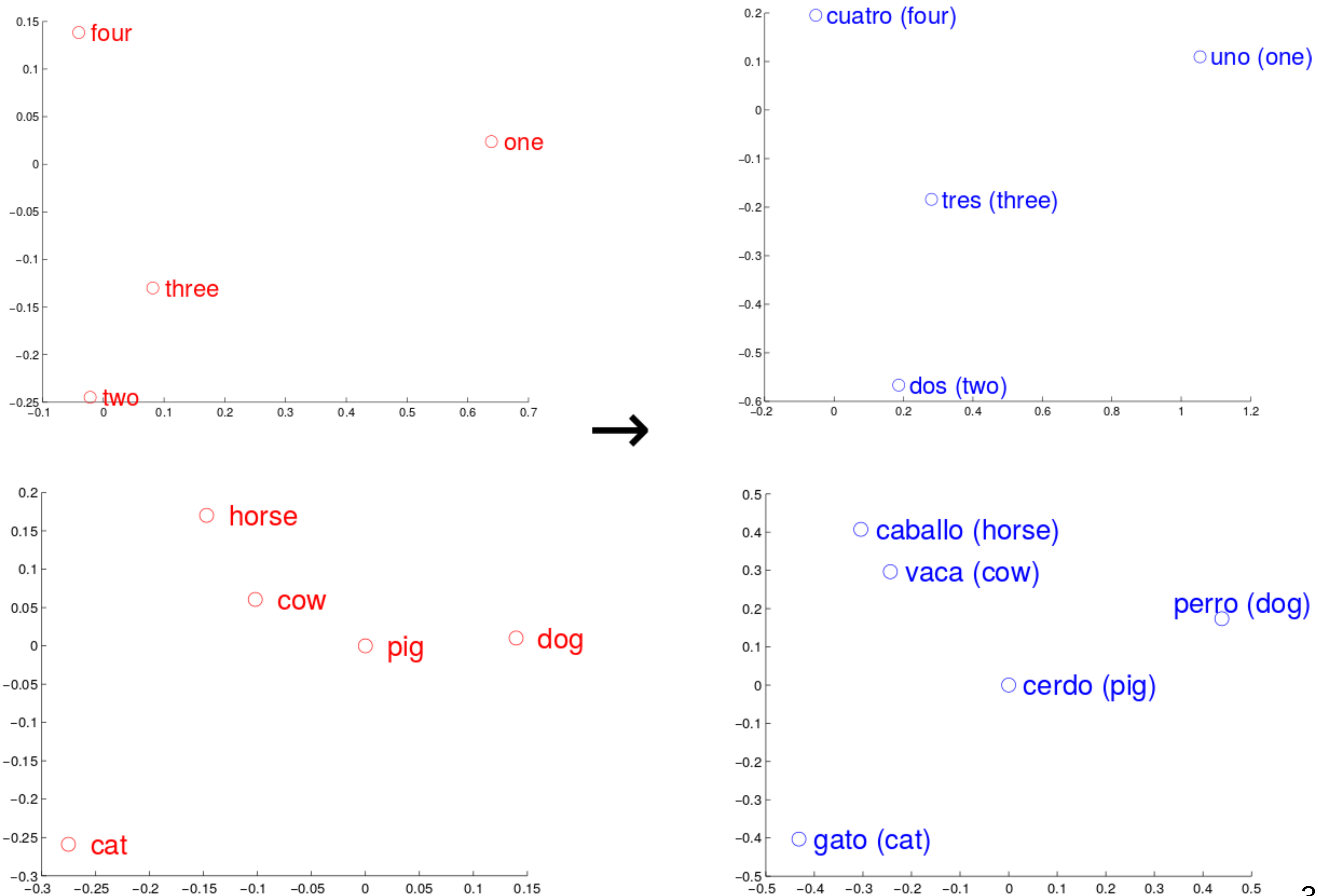
CBOW vs Skip-gram



CBOW:	Faster to train than skip-gram, slightly better accuracy for frequent words
Skip-gram:	Works well with small amount of data, represents well even rare words or phrases

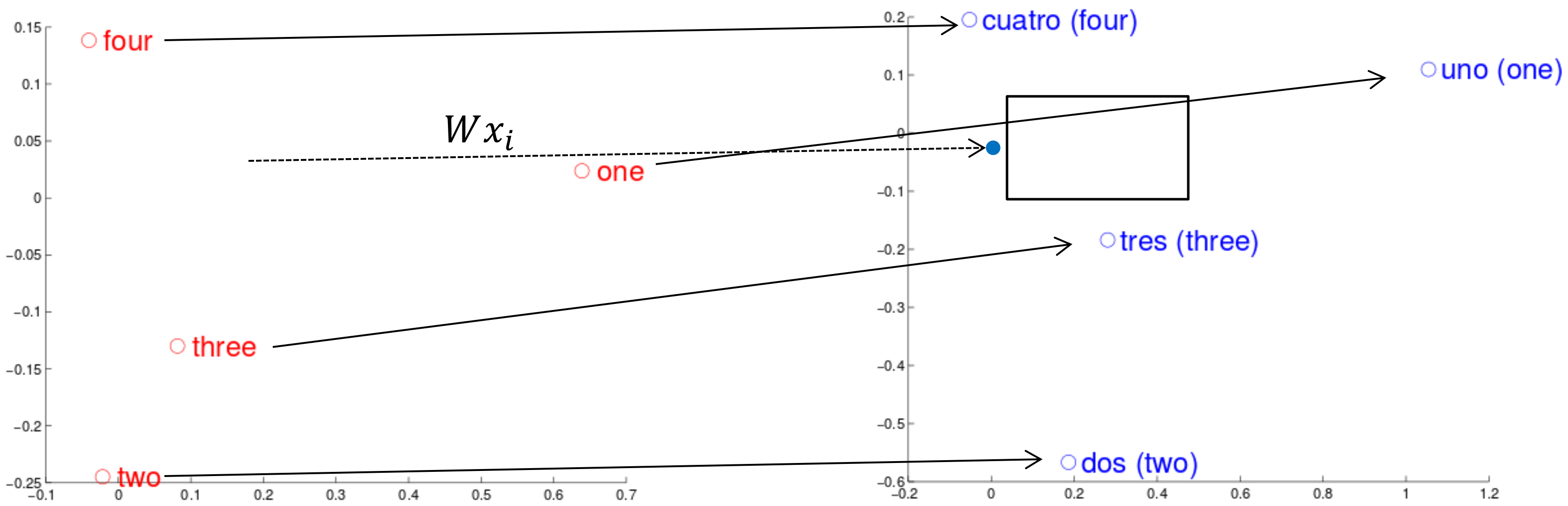
[Mikolov 2013]

word2vec for Machine Translation



[Mikolov 2013]

word2vec for Machine Translation



1. word2vec on EN+ES language corpora
2. Learn mapping between spaces (translation matrix W)
3. Map new input: $Wx_i = z_i$
4. Choose word with closest distance to z_i

[Mikolov 2013]

word2vec for Machine Translation

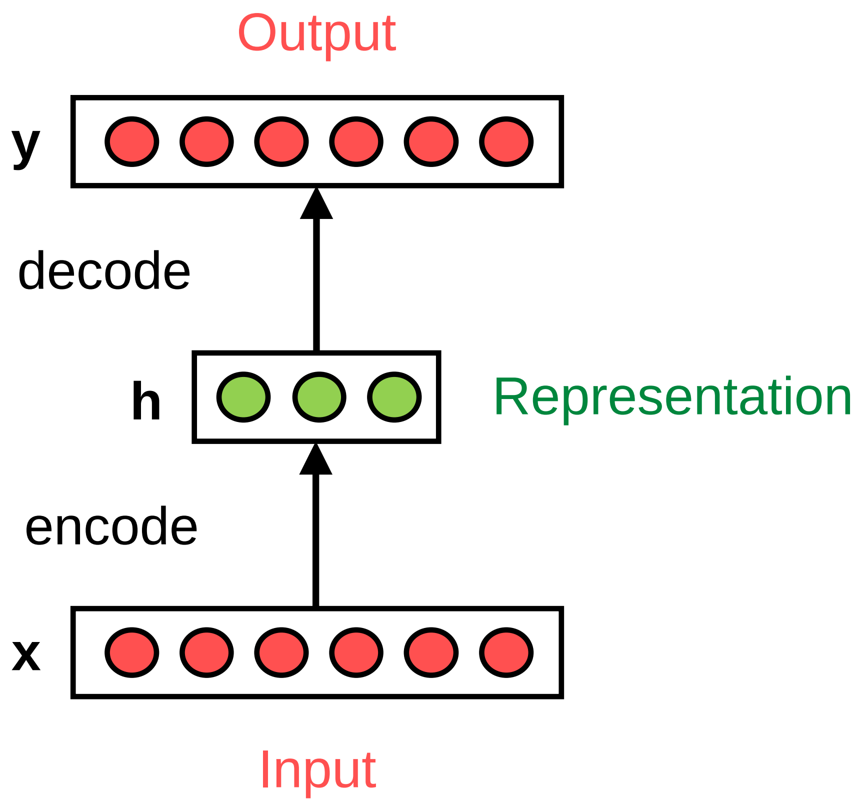
Spanish word	Computed English Translations	Dictionary Entry
emociones	emotions emotion feelings	emotions
protegida	wetland undevelopable protected	protected
imperio	dictatorship imperialism tyranny	empire
determinante	crucial key important	determinant
preparada	prepared ready prepare	prepared
millas	kilometers kilometres miles	miles

[Mikolov 2013]

From Autoassociator / Autoencoder to RAAM

- Autoassociator: Special case of MLP, as in word2vec
- Dimensionality reduction, representation learning

- Trained to reconstruct own input:
  - input = target
  - $|\mathbf{x}| = |\mathbf{y}|$
    $|\mathbf{h}| < |\mathbf{x}|$
- unsupervised
  (often used for pretraining)

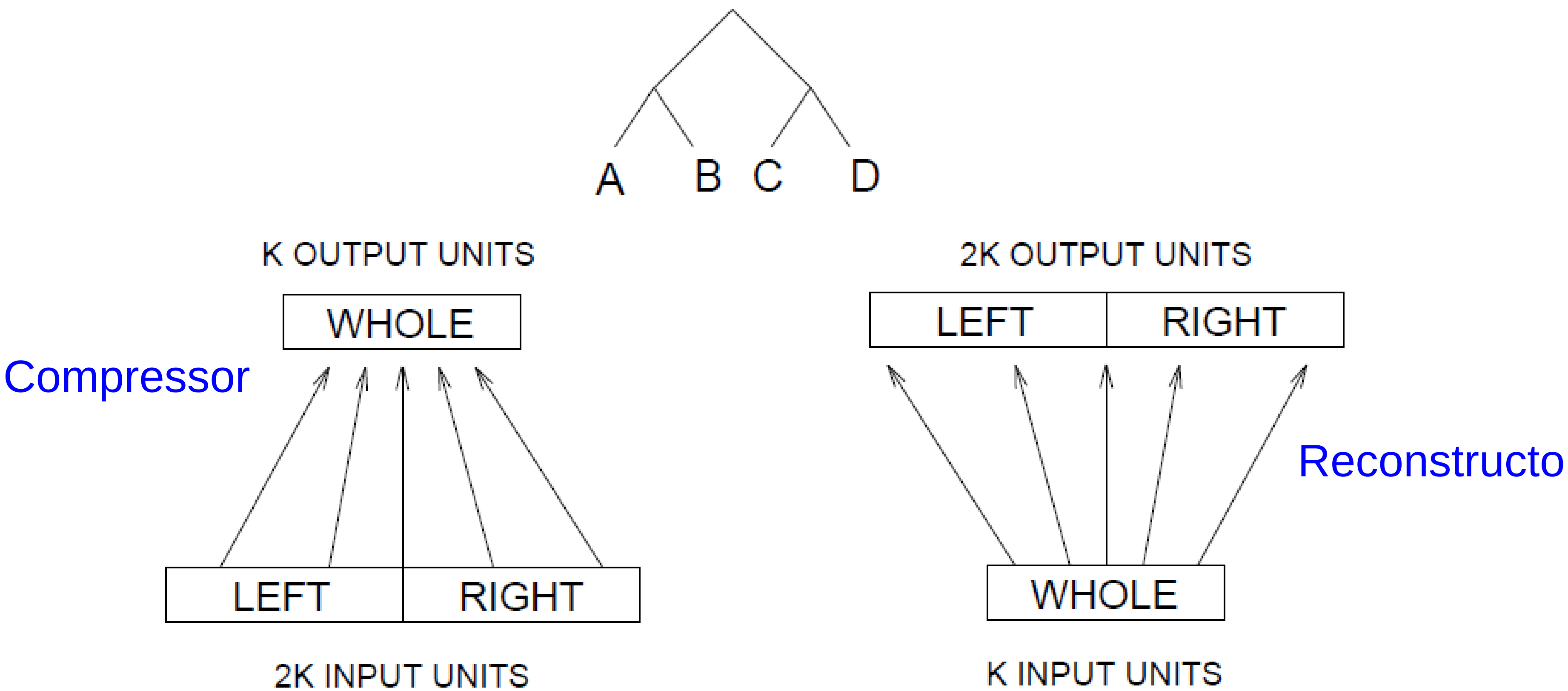


RAAM: Recursive Autoassociative Memory

- Compressor (encoder) maps input to internal reduced representation
- Reconstructor (decoder) decodes the internal representation into output
- Trained and applied recursively
- Can represent tree-like structures:

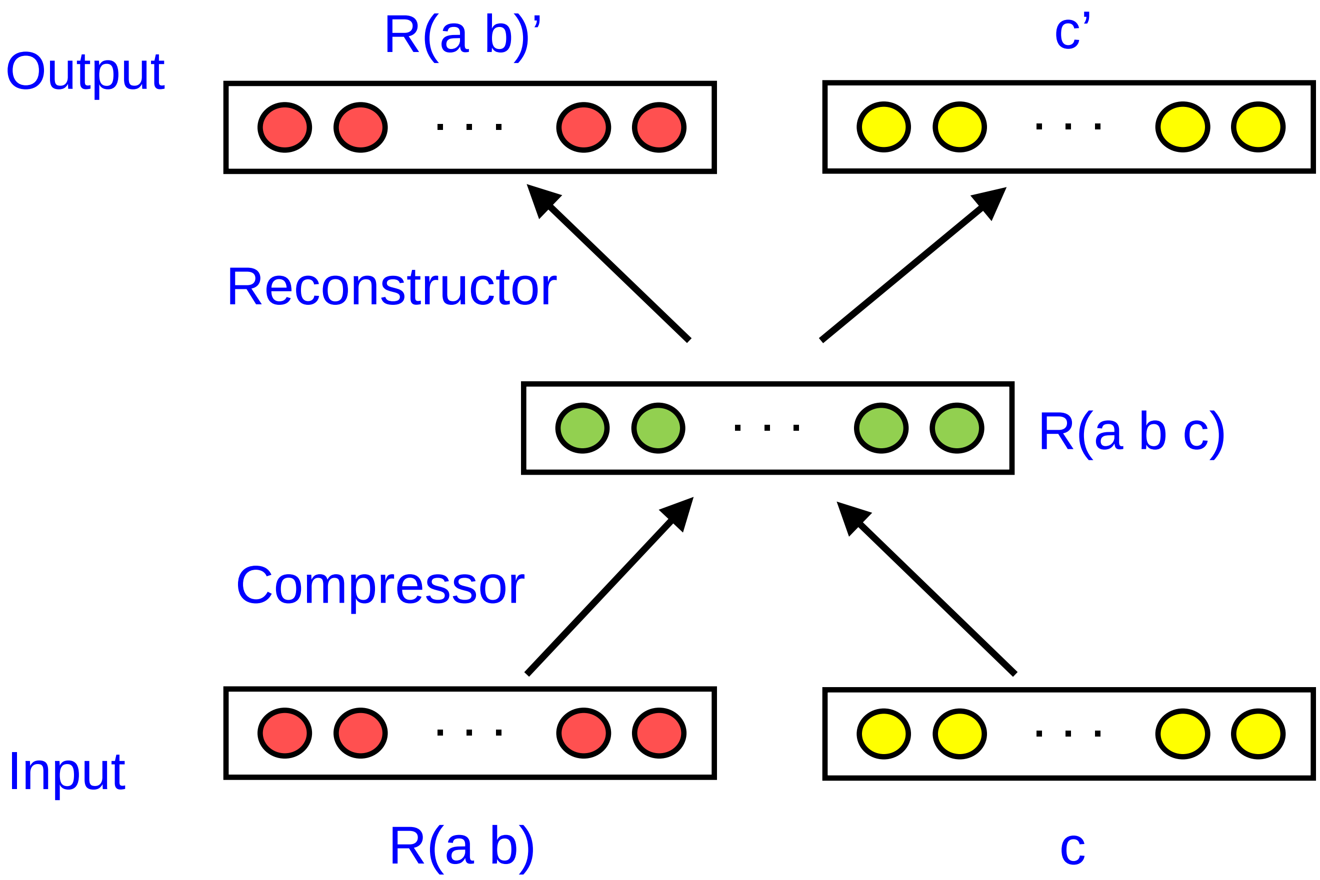
- (det (adj noun)) trained as
  - (adj noun)-> R(adj noun) -> (adj* noun*)
  - (det R(adj noun)) -> R(det adj noun) -> (det* R(adj noun)*)

Compressor and Reconstructor of binary trees

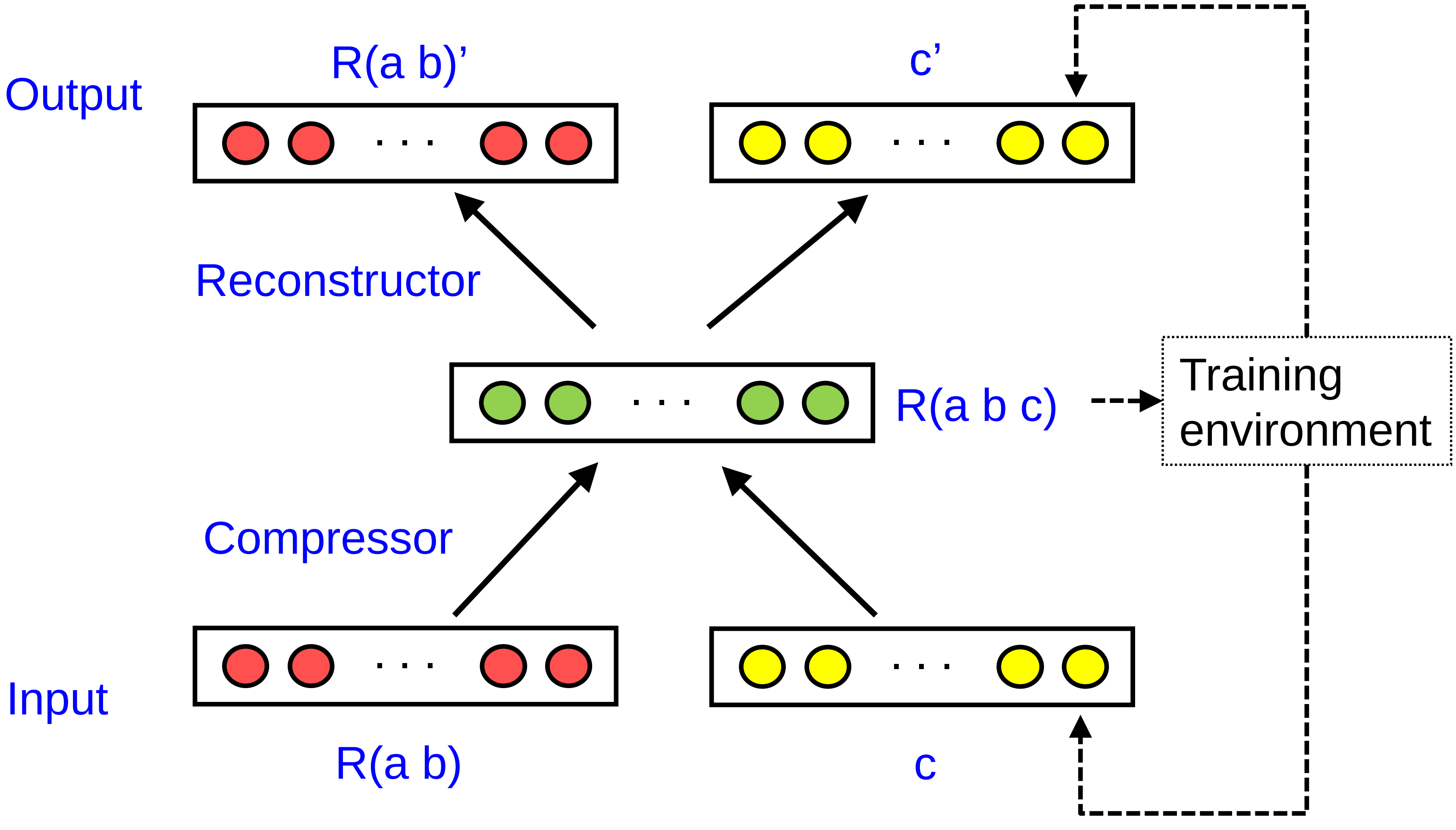


input pattern		*hidden pattern*		*output pattern*
$(A \quad B)$	$\rightarrow$	$R_1(t)$	$\rightarrow$	$(A'(t) \quad B'(t))$
$(C \quad D)$	$\rightarrow$	$R_2(t)$	$\rightarrow$	$(C'(t) \quad D'(t))$
$(R_1(t) \quad R_2(t))$	$\rightarrow$	$R_3(t)$	$\rightarrow$	$(R_1(t)' \quad R_2(t)')$

RAAM
Recursive autoassociative memory



RAAM
Recursive autoassociative memory



(Pollack 90)

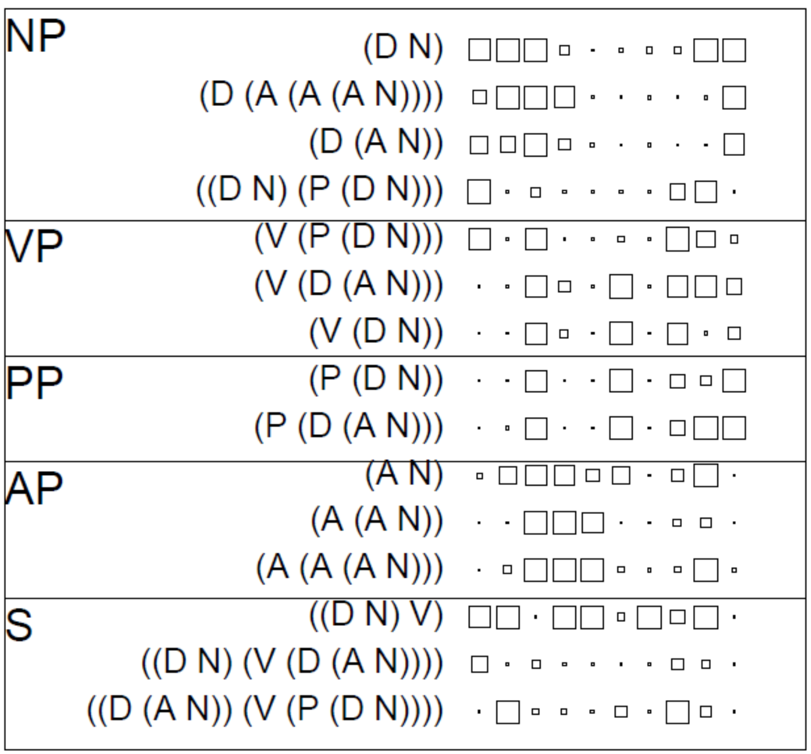
RAAM
Learning representations for syntactic trees

Grammar

S -> NP VP | NP V
NP -> D AP | D N | NP PP
PP -> P NP
VP -> V NP | V PP
AP -> A AP | A N

Sequences

(D (A (A (A N))))
((D N)(P (D N)))
(V (D N))
(P (D (A N)))
((D N) V)
((D N) (V (D (A N))))
((D (A N)) (V (P (D N))))



Learned Internal Representations

Summary

- Sequence learning on character- and word-level
  - char-rnn, SCREEN
- Distributed word representations for sequence learning
  - RAAM, word2vec
  - Visualization of hidden layers (t-SNE, hierarchical clustering)

- Further reading:
  Optional research papers at Commsy / Knowledge Technology website
  http://www.informatik.uni-hamburg.de/WTM/