LISA

Université de Montréal

# Learning to Represent Semantics

**Yoshua Bengio**

Words2Actions Workshop,

NAACL HLT 2012, Montreal

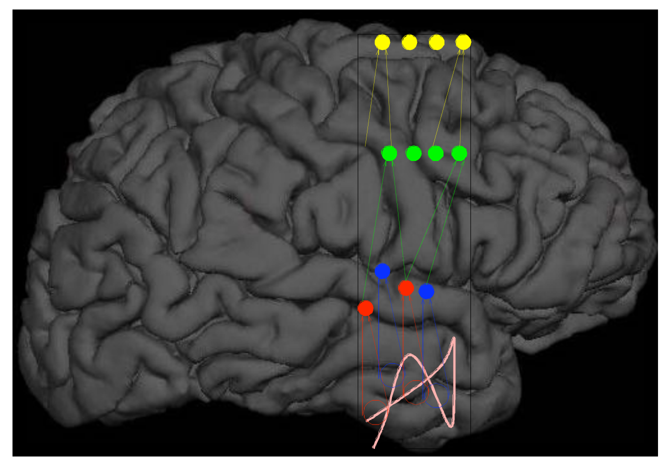
# From AI to Deep Learning

- AI requires operational knowledge
- Handcrafting it all is daunting, brittle, incomplete, failed: **learn it**
- Most common now: hand-crafted features + simple (linear) ML
- Without the right (task-specific) features: curse of dimensionality
- Need for learning the features: representation-learning
- Theoretical and empirical evidence in favor of multiple levels of representation (Deep Learning)

# Deep Learning: General Motivation

- Learning features
  - Learn features as part of a machine learning system
  - Not all features can be  explicitly described by experts

- Biologically inspired learning
  - Brain has a deep architecture
  - Cortex seems to have a generic learning algorithm
  - **Humans first learn simpler concepts and then compose them to more complex ones**



# Deep Learning: General Motivation

- It works well already for vision, NLP, collaborative filtering,…
- Wins two transfer learning competitions in 2011
- State of the art performance for POS, NER, Chunking

| Task | | Benchmark | SENNA |
|---|---|---|---|
| Part of Speech (POS) | (Accuracy) | 97.24 % | 97.29 % |
| Chunking (CHUNK) | (F1) | 94.29 % | 94.32 % |
| Named Entity Recognition (NER) | (F1) | 89.31 % | 89.59 % |

(Collobert et al., 2011)

- Sentiment analysis on opinions, experiences, movies
- Paraphrase detection (Socher et al. 2011)
- Relation classification
- Language Modeling (Schwenk et al, Mikolov et al)

# Deep Learning Motivation for Semantics

- Language Models: model joint probability of word sequences

- Training sentence

  *The cat is walking in the bedroom*

- Test sentence:

  *A dog was running in a room*

- Sparsity / curse of dim. problem for longer n-grams
- Possible Solutions: back-off, word classes (too coarse)
- Better: similar representations for semantically similar phrases

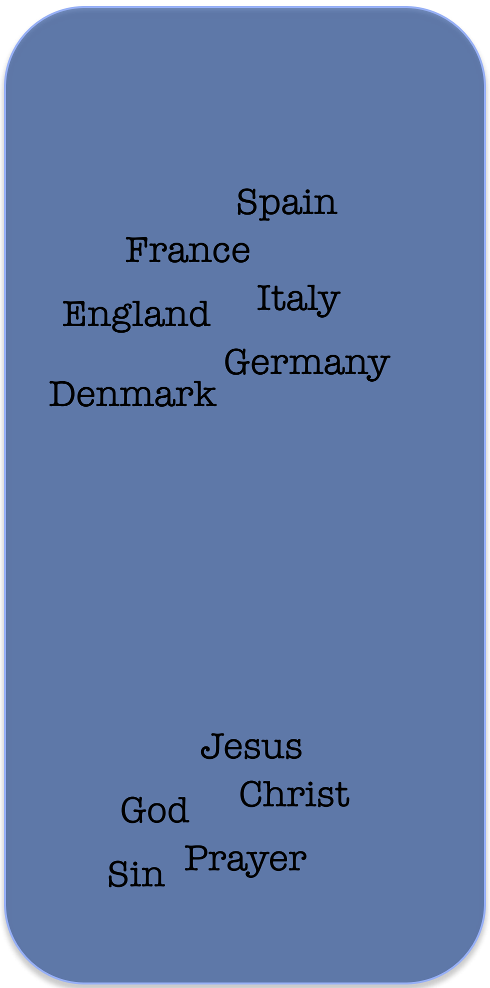
# 1st step: represent words

- Deep learning can learn a distributed continuous-valued vector for each word from raw text:


| France | Jesus | XBOX | Reddish | Scratched |
|---|---|---|---|---|
| Spain | Christ | Playstation | Yellowish | Smashed |
| Italy | God | Dreamcast | Greenish | Ripped |
| Russia | Resurrection | PS### | Brownish | Brushed |
| Poland | Prayer | SNES | Bluish | Hurled |
| England | Yahweh | WH | Creamy | Grabbed |
| Denmark | Josephus | NES | Whitish | Tossed |
| Germany | Moses | Nintendo | Blackish | Squeezed |
| Portugal | Sin | Gamecube | Silvery | Blasted |
| Sweden | Heaven | PSP | Greyish | Tangled |
| Austria | Salvation | Amiga | Paler | Slashed |

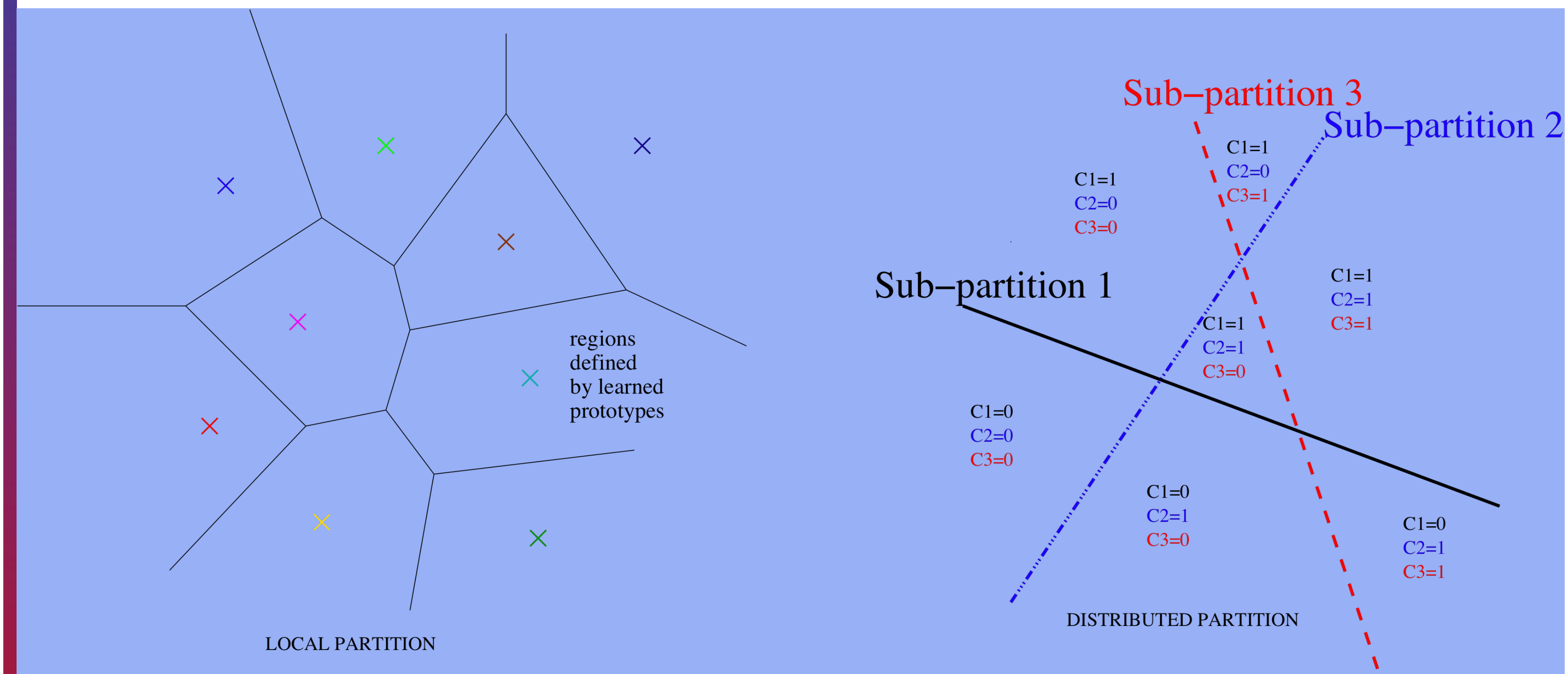
*Collobert & Weston, ICML'2008*

# Distributed Representations




- In contrast to the the "atomic" or "localist" representations employed in traditional cognitive science, a distributed representation is one in which "each entity is represented by a pattern of activity distributed over many computing elements, and each computing element is involved in representing many different entities".
- Hinton (1984) "Distributed representations" CMU-CS-84-157

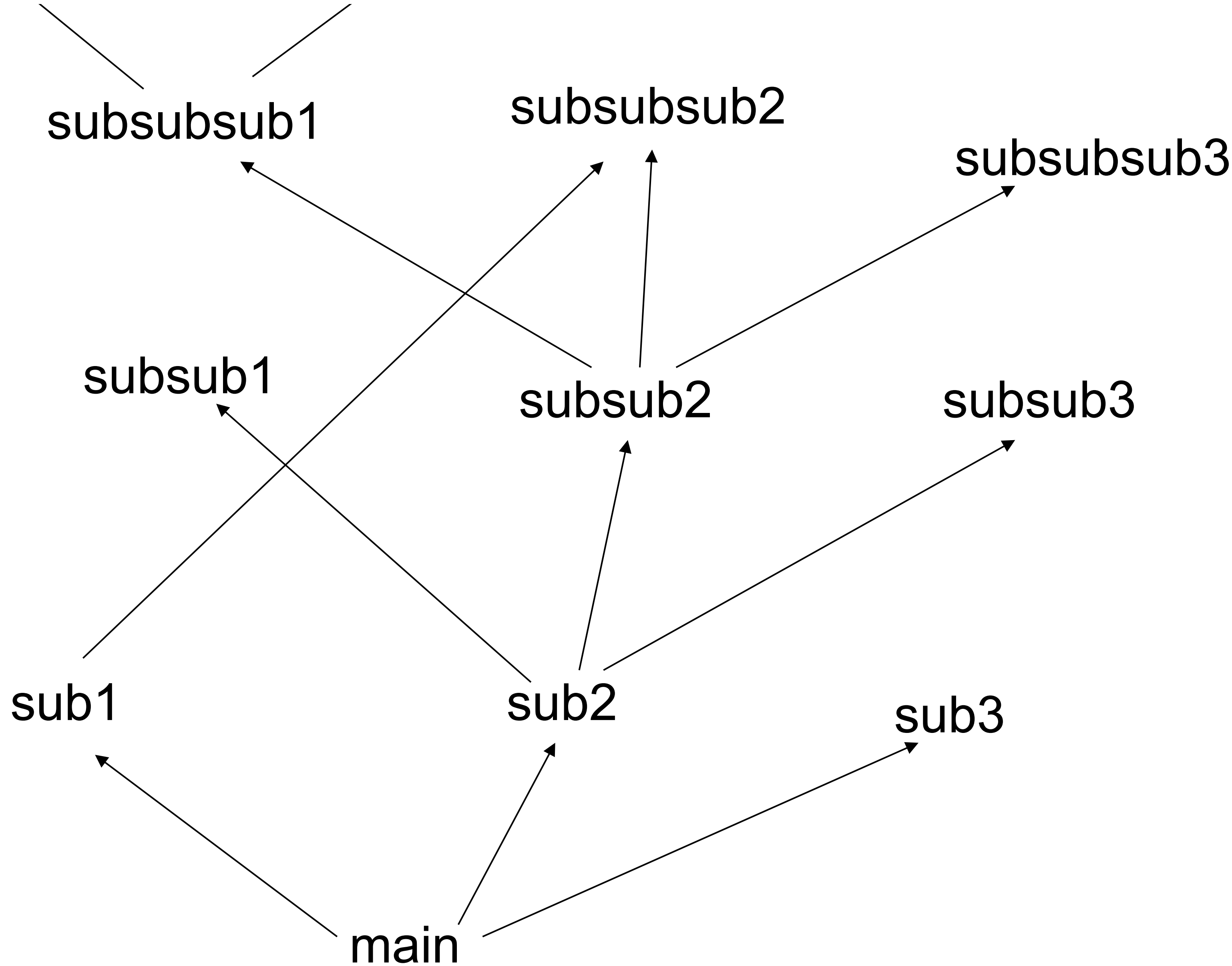
# Local vs Distributed Latent Variables/Attributes




Clustering

Multi-clustering

## 2nd step: learn to compose words into phrases and semantic relations




**“Deep” computer program**

subroutine1 includes
subsub1 code and
subsub2 code and
subsubsub1 code

subroutine2 includes
subsub2 code and
subsub3 code and
subsubsub3 code and …

main

**“Shallow” computer program**

# "Deep" circuit

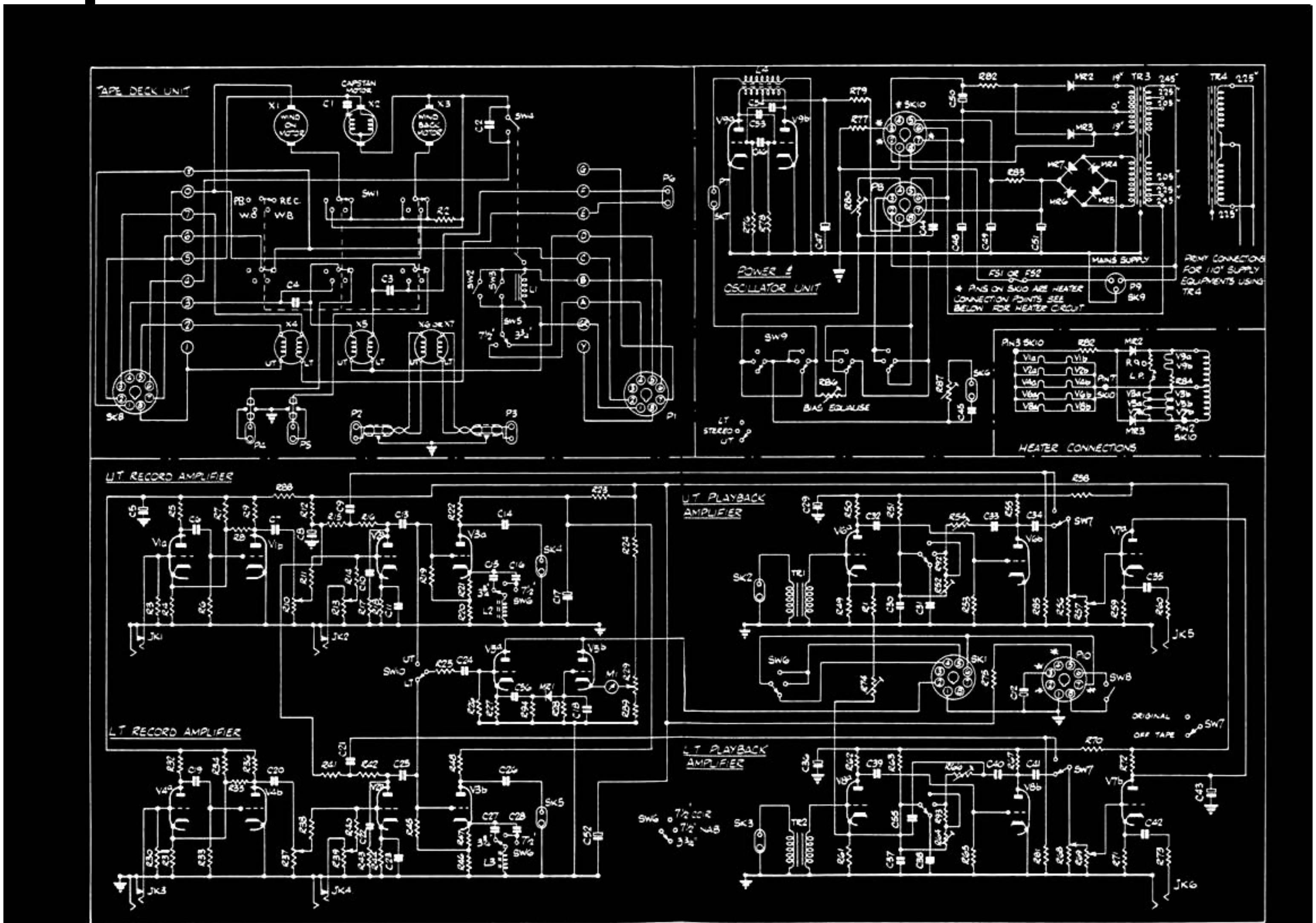
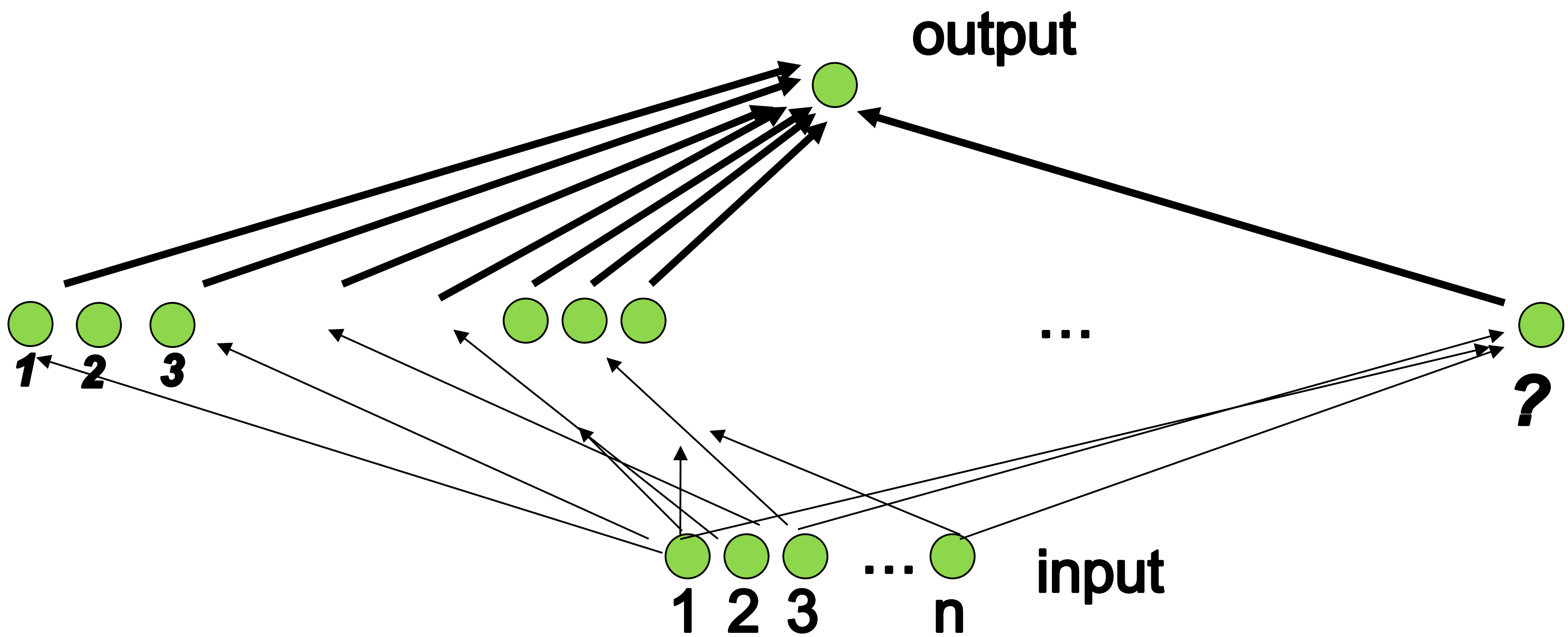



FIG. 16. COMPLETE CIRCUIT DIAGRAM, SERIES 420

# “Shallow” circuit




Falsely reassuring theorems: one can approximate any reasonable (smooth, boolean, etc.) function with a 2-layer architecture

# Deep Architectures are More Expressive

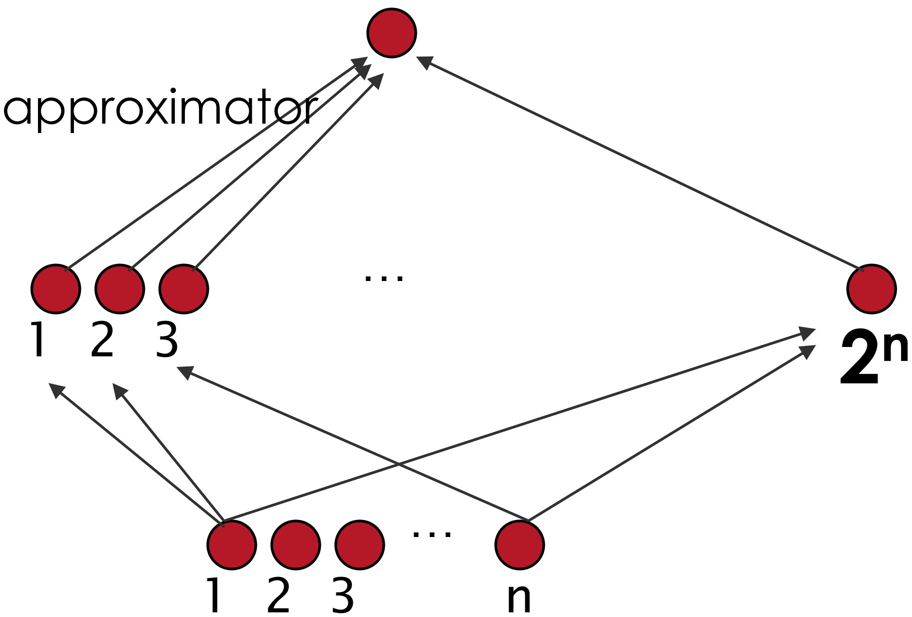
Theoretical arguments:

2 layers of { Logic gates / Formal neurons / RBF units } = universal approximator

RBMs & auto-encoders = universal approximator

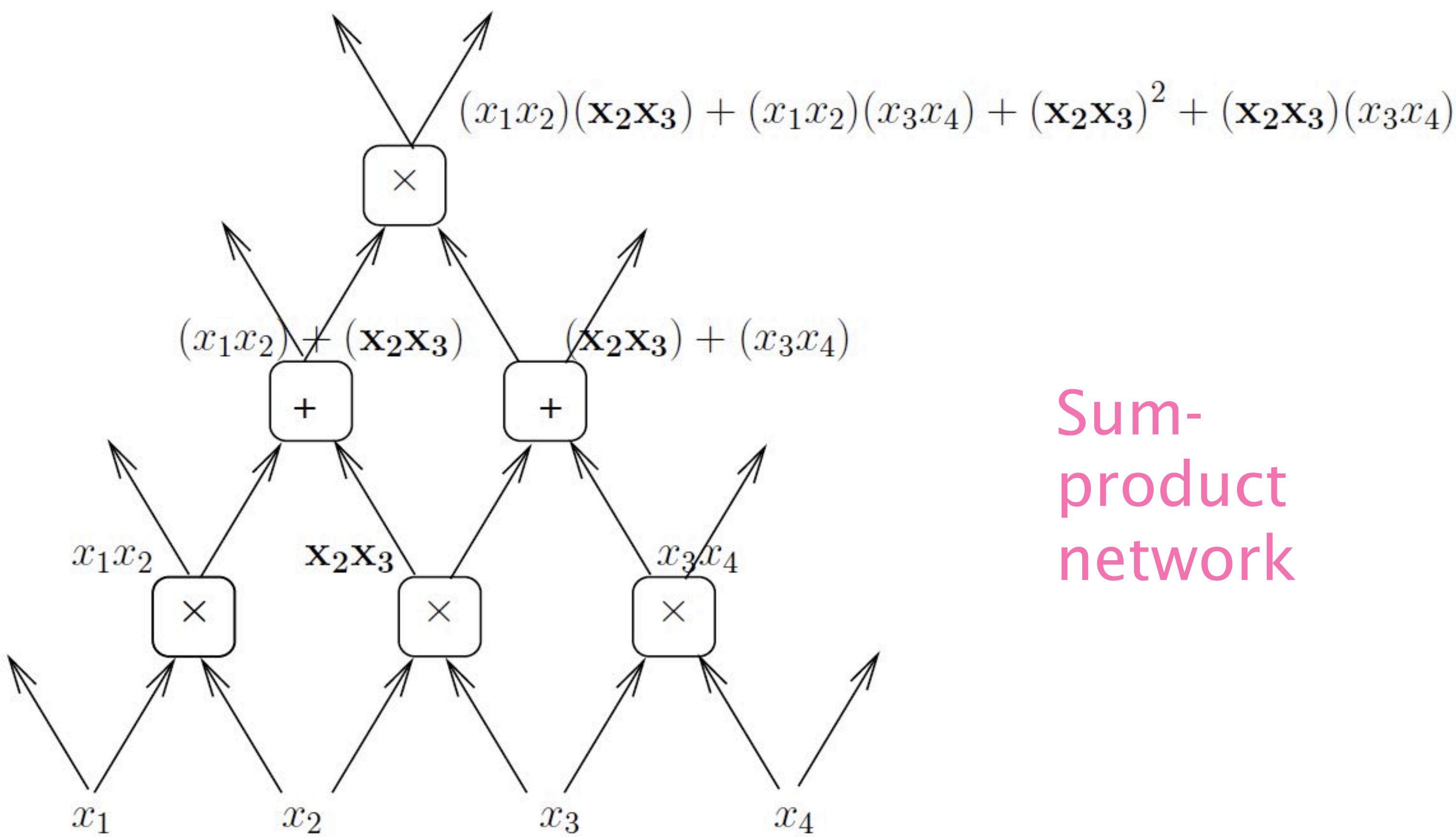
Theorems on advantage of depth:
(Hastad et al 86 & 91, Bengio et al 2007, Bengio & Delalleau 2011, Braverman 2011)

Functions compactly represented with k layers may require exponential size with 2 layers



# Sharing Components in a Deep Architecture

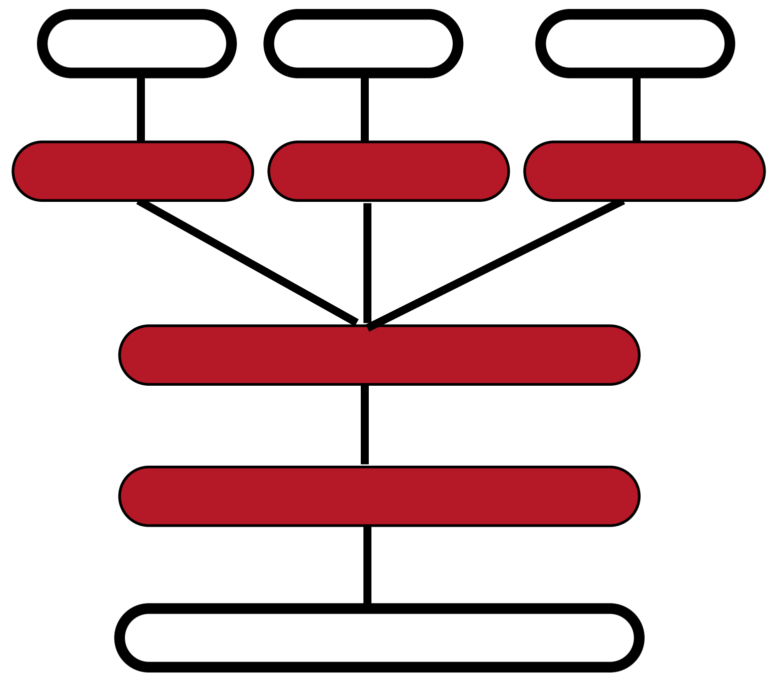
Polynomial expressed with shared components: advantage of depth may grow exponentially



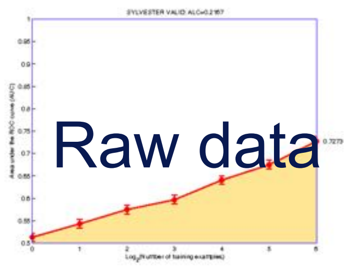

Sum-product network

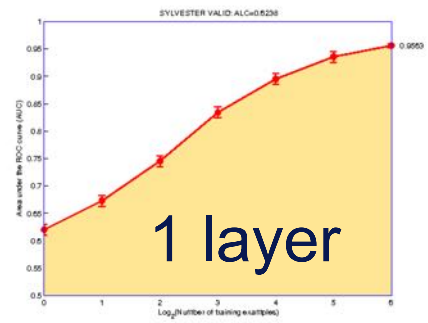
# Deep Architectures and Sharing Statistical Strength, Multi-Task / Transfer Learning

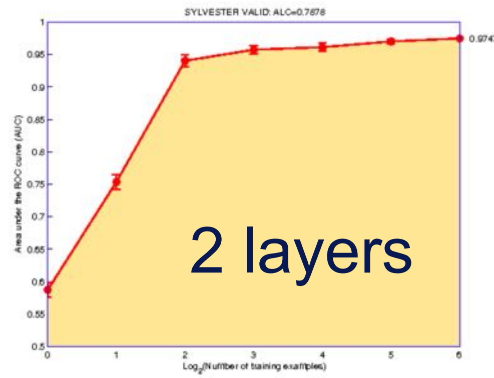
- Generalizing better to new tasks & domains is crucial to approach AI
- Deep architectures can learn good intermediate representations shared across tasks
- Good representations are often those making sense for many tasks because they capture underlying factors = semantics



# Unsupervised and Transfer Learning Challenge + Transfer Learning Challenge: Deep Learning 1st Place

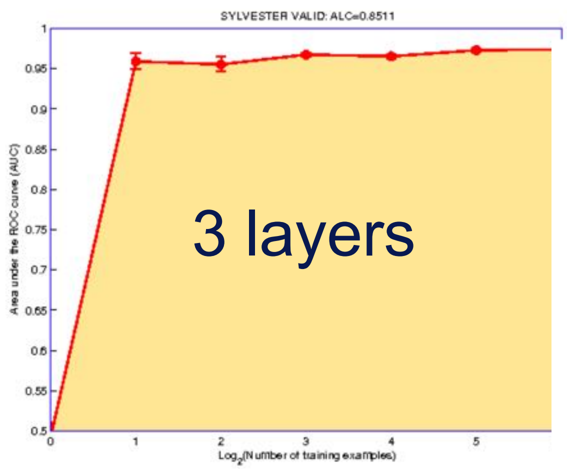


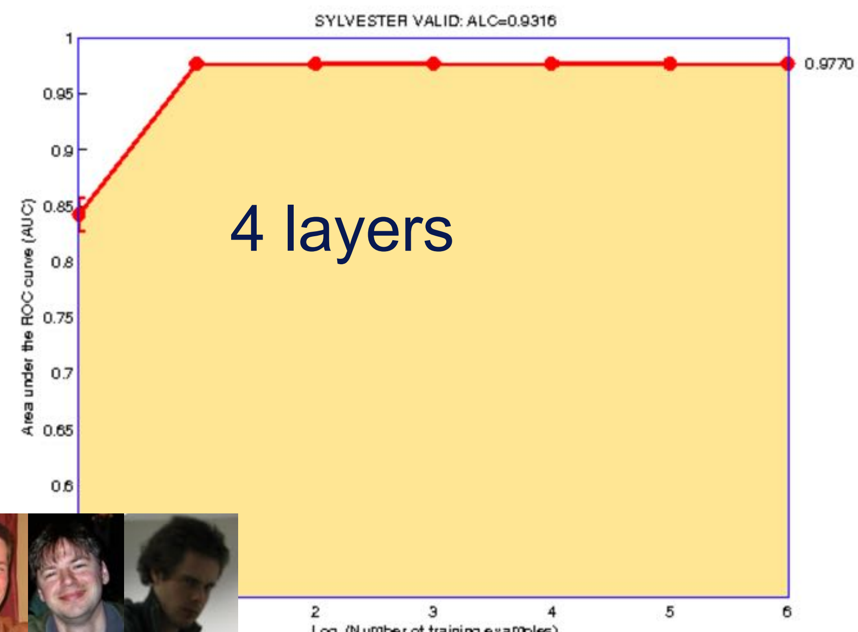




NIPS'2011 Transfer Learning Challenge
Paper: ICML'2012

ICML'2011 workshop on Unsup. & Transfer Learning




## Invariance and Disentangling

- Invariant features

- Which invariances?

- Alternative: learning to disentangle factors

- Good disentangling →

  avoid the curse of dimensionality

# Advantages of Sparse Representations

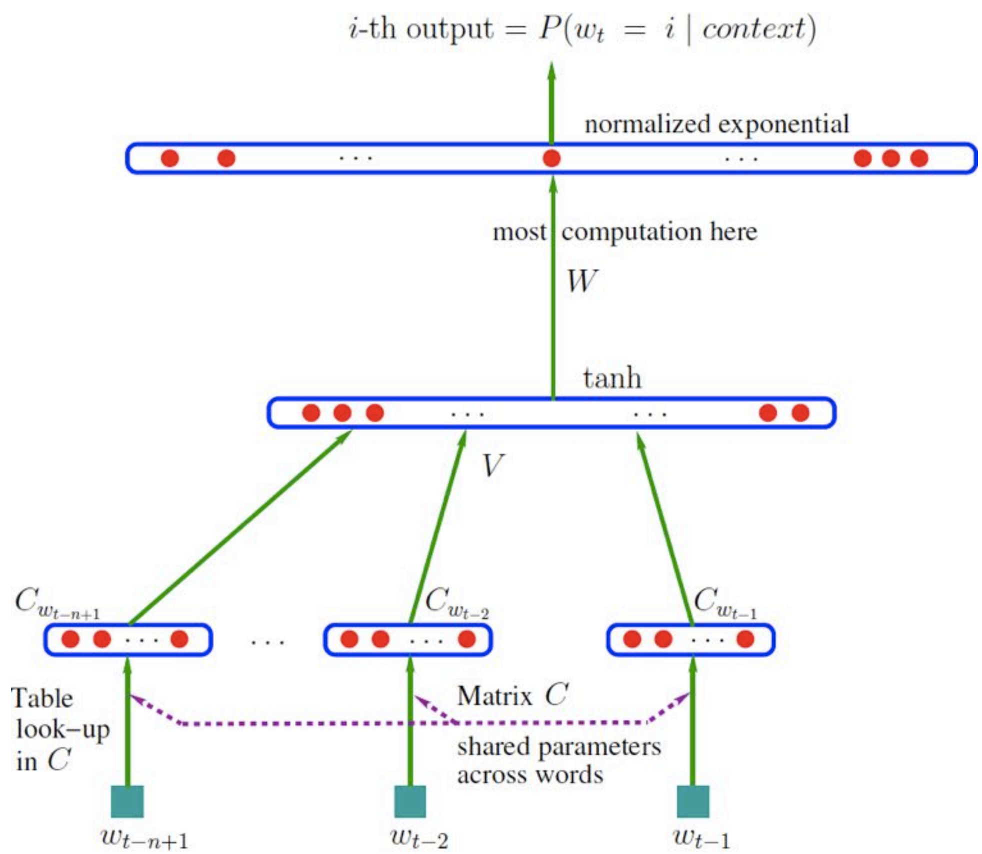
- Just add a penalty on learned representation
- Information disentangling (compare to dense compression)
- More likely to be linearly separable (high-dimensional space)
- Locally low-dimensional representation = local chart
- Hi-dim. sparse = efficient variable size representation
  = data structure

Few bits of information

Many bits of information

# Deep & Distributed NLP

- See "*Neural Net Language Models*" ***Scholarpedia*** *entry*
- *NIPS'2000 and JMLR 2003 "A Neural Probabilistic Language Model"*
  - Each word represented by a distributed continuous-valued code
  - Generalizes to sequences of words that are semantically similar to training sequences



# Deep Learning: Motivations for NLP

- Allows to generalize to sequences of words that are semantically similar to training sequences

- Training sentence

  The cat is walking in the bedroom

- Can generalize to

  A dog was running in a room

- Because of the similarity between distributed representations for (a,the), (cat,dog), (is,was), etc.

## Neural Networks for Learning Word Vectors

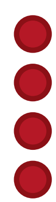
- Idea: A word and its context is a positive training sample, a random word in that same context is a negative training sample:

- cat chills on a mat      cat chills Jeju a mat

- Similar: Implicit negative evidence in Contrastive Estimation, Smith and Eisner (2005)

# A neural network for learning word vectors

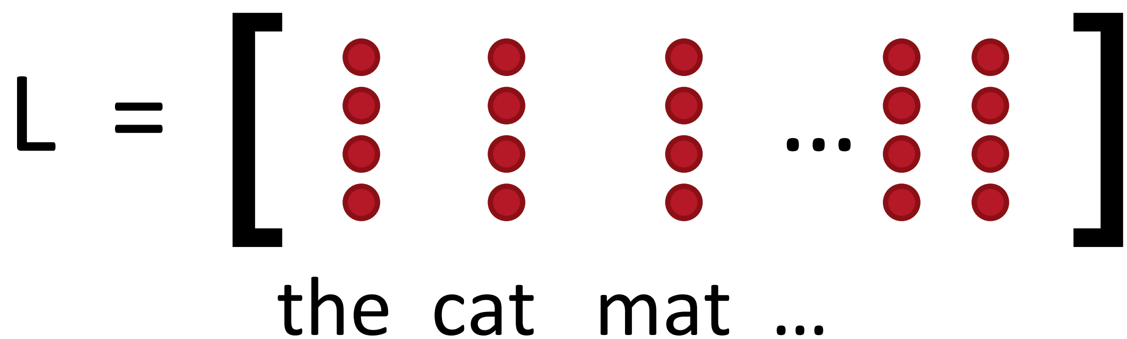
- Idea: A word and its context is a positive training sample, a random word in that same context is a negative training sample.
- score(cat chills on a mat) > score(cat chills Jeju a mat)
- How to compute the score?
  - With a neural network
  - Each word is associated with an n-dimensional vector



## Word embedding matrix

$$L \in \mathbb{R}^{n \times |V|}$$

- Initialize all word vectors randomly to form a word embedding matrix

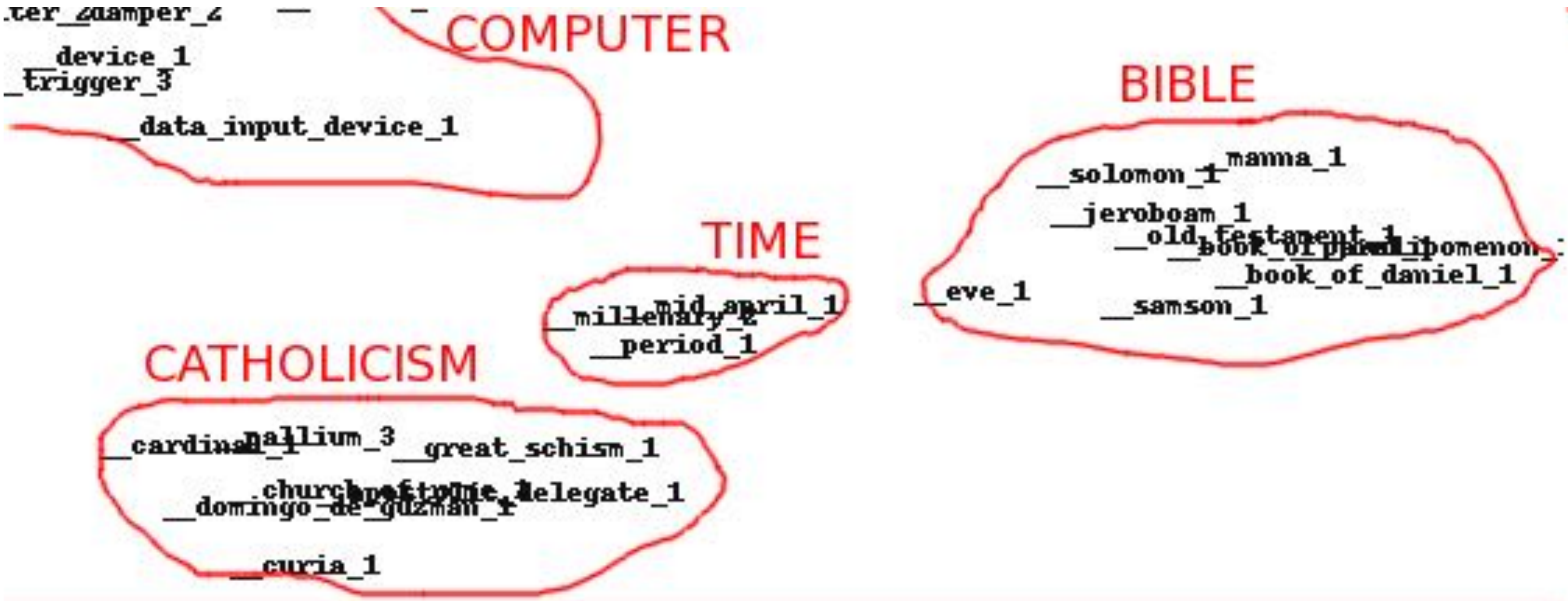



- These are the word features we want to learn
- Also called look-up table

# t-SNE of Embeddings: zoom 1

# t-SNE of Embeddings: zoom 2

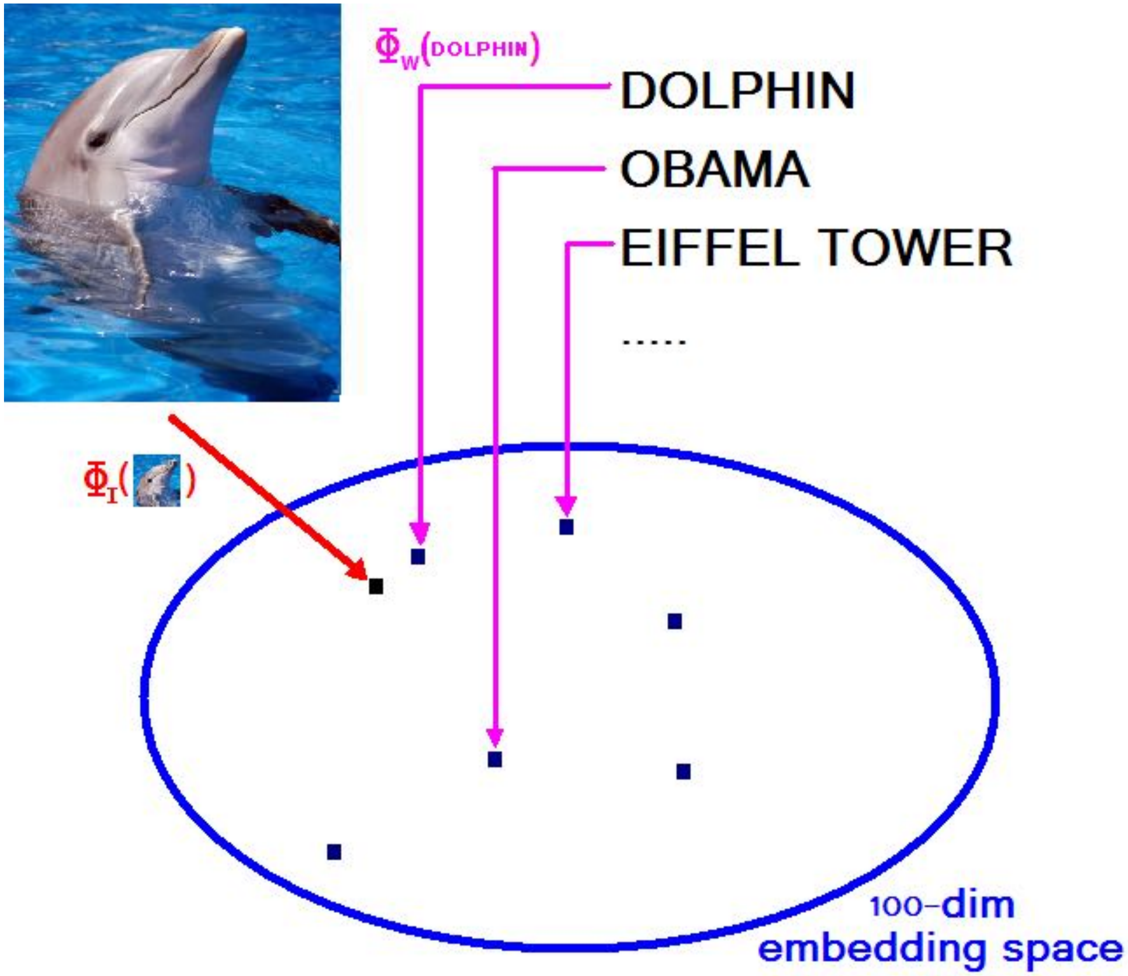
# t-SNE of Embeddings: zoom 3



# Joint Image-Query Embedding Space

S. Bengio, J. Weston et al @ Google

(NIPS'2010, JMLR 2010, MLJ 2010, NIPS'2009)




*Learn* $\Phi_I(\cdot)$ *and* $\Phi_W(\cdot)$ *to optimize precision@k.*

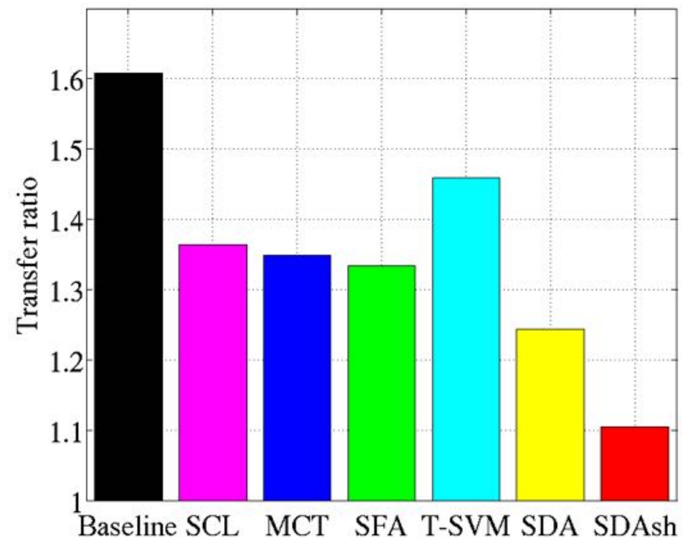
# Some results with deep distributed representations for NLP

- *(Bengio et al 2001, 2003)*: beating n-grams on small datasets (Brown & APNews), but much slower
- *(Schwenk et al 2002,2004,2006)*: beating state-of-the-art large-vocabulary speech recognizer using deep & distributed NLP model, with ***real-time*** speech recognition
- *(Morin & Bengio 2005, Blitzer et al 2005, Mnih & Hinton 2007,2009)*: better & faster models through hierarchical representations
- *(Collobert & Weston 2008)*: reaching state-of-the-art in multiple NLP tasks (**SRL**, POS, NER, chunking) thanks to unsupervised pre-training and multi-task learning
- *(Bai et al 2009)*: ranking & semantic indexing (info retrieval).
- *(Collobert 2010):* Deep Learning for Efficient Discriminative Parsing
- (S. Bengio, J. Weston et al @ Google, 2009,2010,2011): joint embedding space for images and keywords, **Google image search**
- *(Sutskever & Martens 2011)*: beating SOA in text compression.
- *(Socher et al 2011)*: parsing with recursive nets, ICML 2011 distinguished application paper award
- *(Mikolov et al 2011)*: beating the SOA in perplexity with recurrence

# Domain Adaptation (ICML 2011)

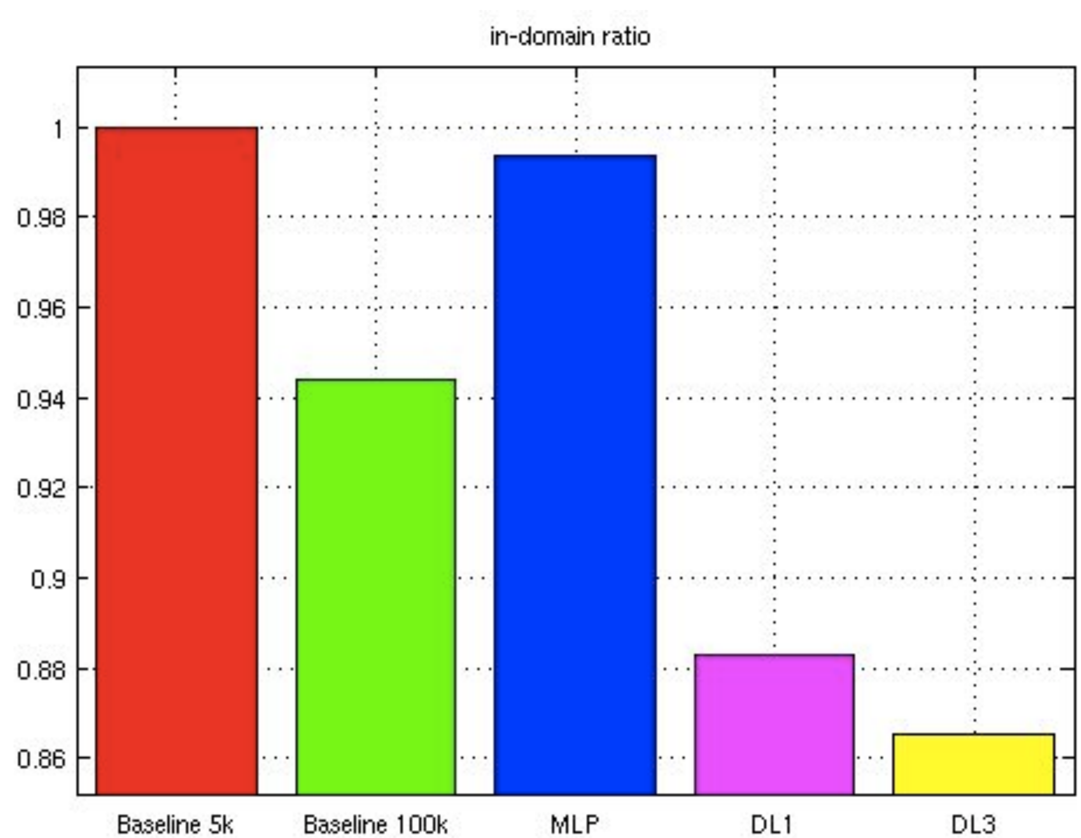
Small (4-domain) Amazon benchmark: we beat the state-of-the-art handsomely

**Sparse rectifiers Stacked Denoising Autoencoders** find more features that tend to be useful either for predicting domain or sentiment, not both = disentangling?

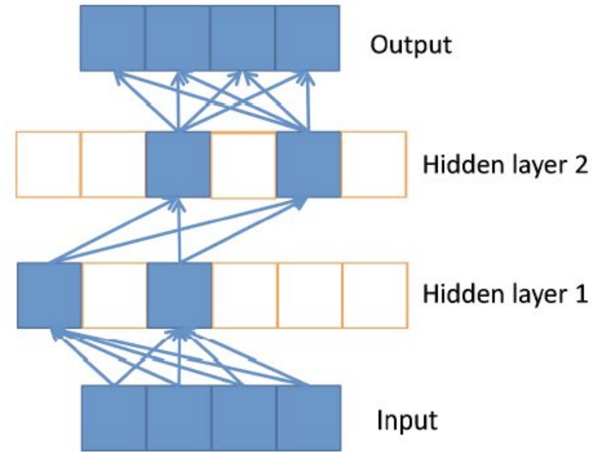


# Sentiment Analysis: Transfer Learning

- 25 Amazon.com domains: toys, software, video, books, music, beauty, …
- Unsupervised pre-training of input space on all domains
- Supervised SVM on 1 domain, generalize out-of-domain
- Baseline: bag-of-words + SVM



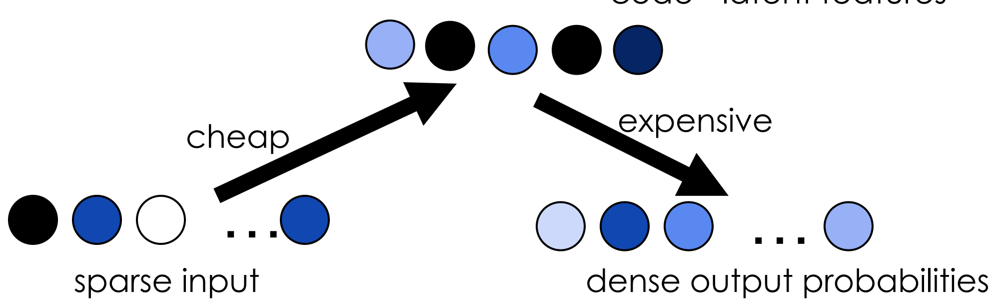
# Representing Sparse High-Dimensional Stuff



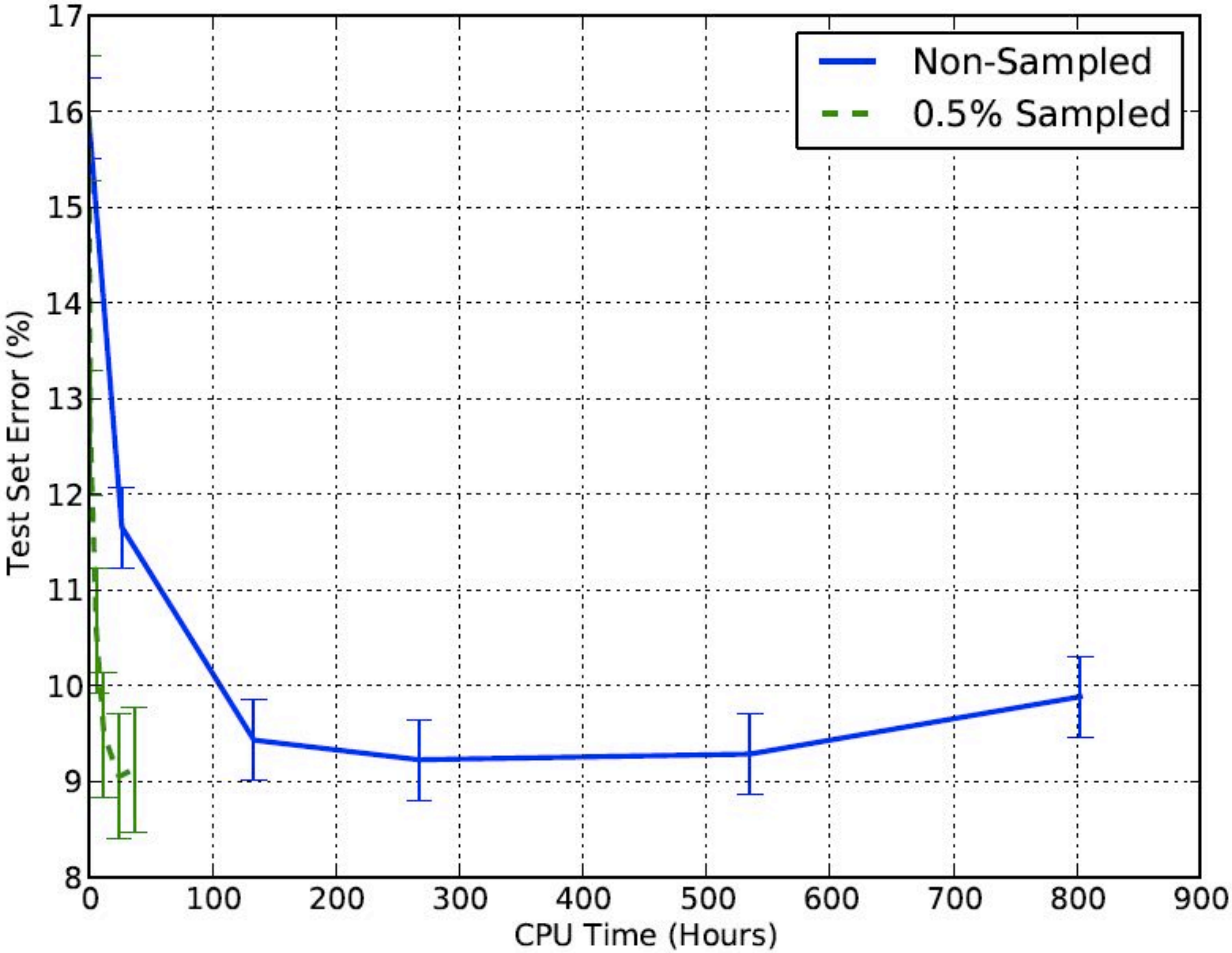

$$f(x) = max(0, x)$$

*Deep Sparse Rectifier Neural Networks*, Glorot, Bordes & Bengio, AISTATS 2011.

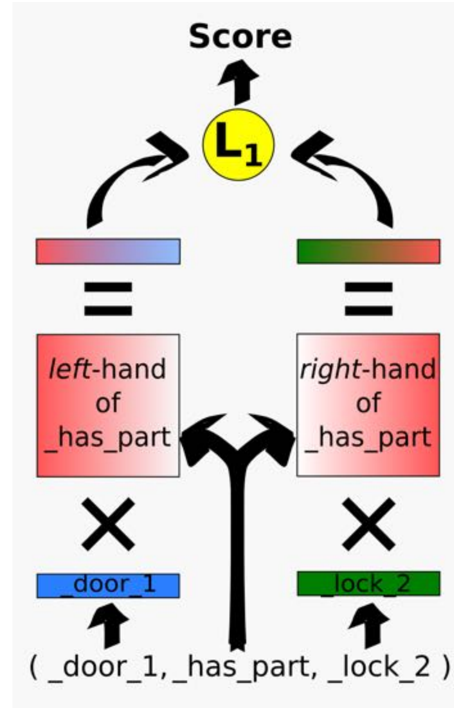
*Sampled Reconstruction for Large-Scale Learning of Embeddings*, Dauphin, Glorot & Bengio, ICML 2011.



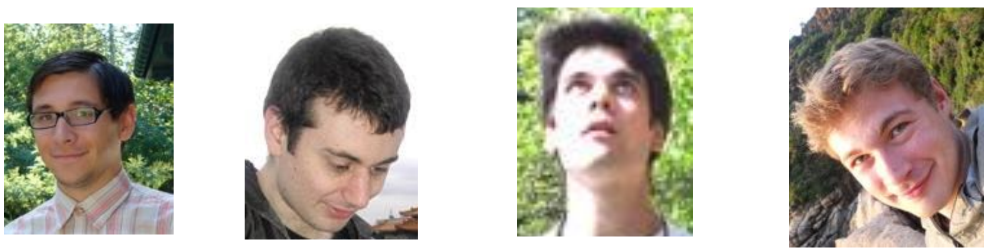
# Speedup from *Sampled Reconstruction*



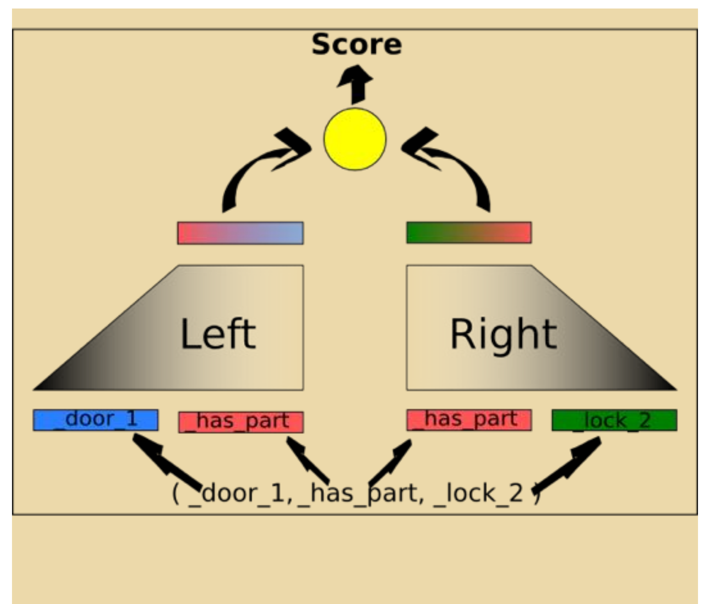
# Modeling Semantics



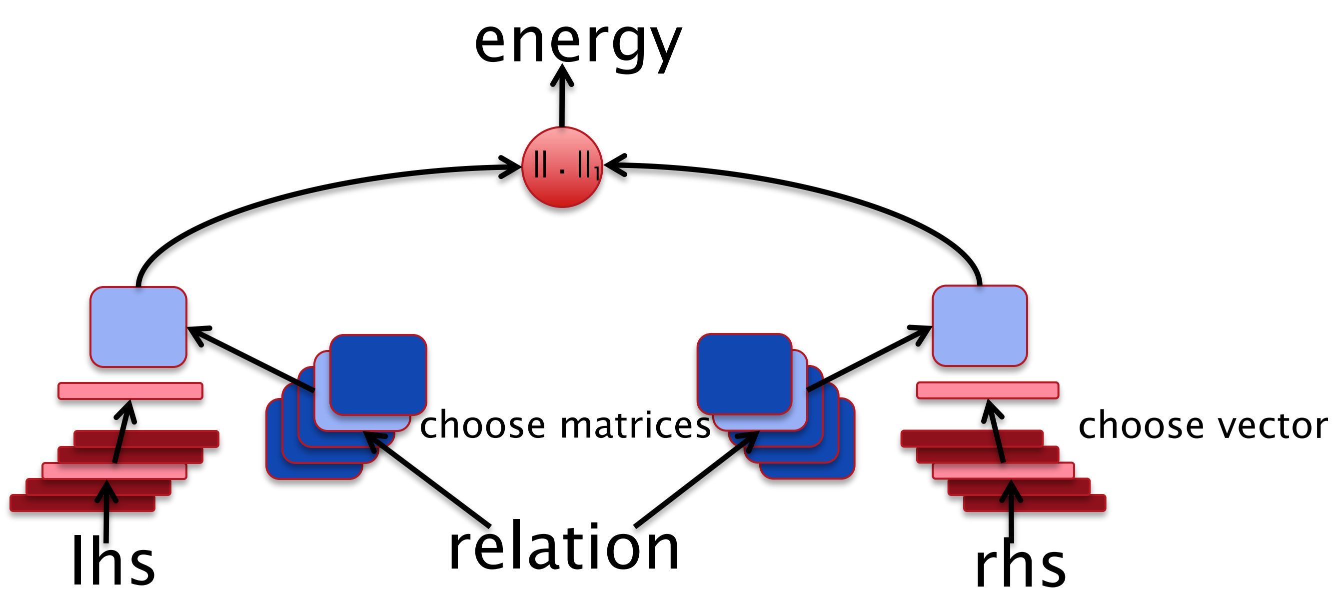

*Learning Structured Embeddings of Knowledge Bases*, Bordes, Weston, Collobert & Bengio, AAAI 2011



*Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing*, Bordes, Glorot, Weston & Bengio, AISTATS 2012

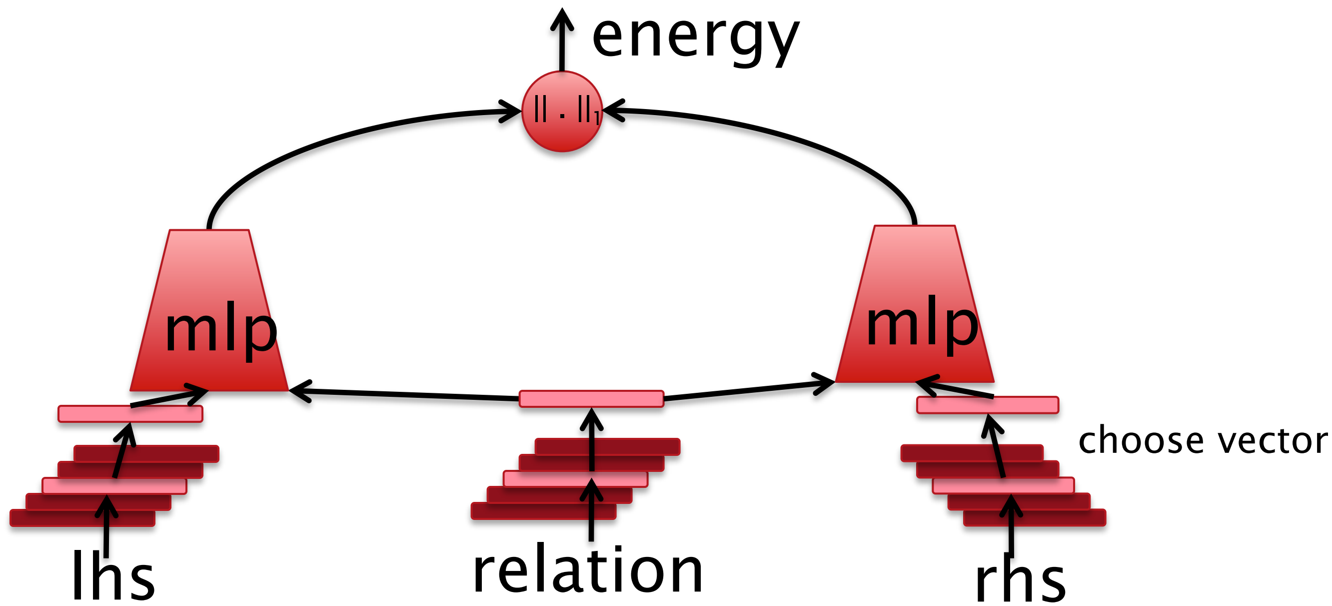


# Modeling Relations with Matrices




Model (lhs, relation, rhs)
Each concept = 1 embedding vector
Each relation = 2 matrices
Ranking criterion
Energy = low for training examples, high o/w

# Allowing Relations on Relations



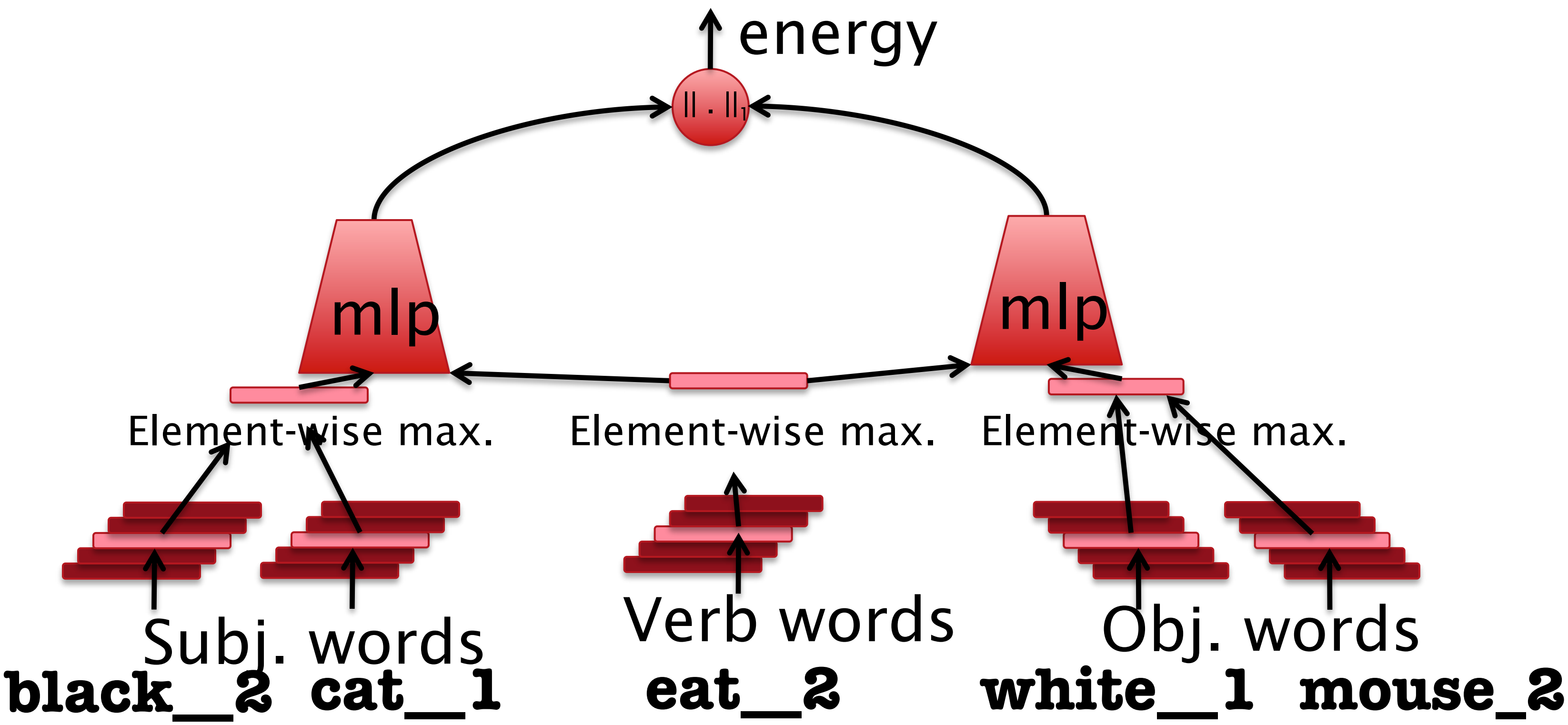

Verb = relation. Too many to have a matrix each.

Each concept = 1 embedding vector

Each relation = 1 embedding vector

Can handle relations on relations on relations

# Training on Full Sentences




→ Use SENNA (Collobert 2010) = embedding-based NLP tagger for Semantic Role Labeling, breaks sentence into
(subject part, verb part, object part)

→ Use max-pooling to aggregate embeddings of words inside each part

# Combining Multiple Sources of Evidence with Shared Embeddings

- The undirected graphical model version of relational learning
- With embeddings (shared representations) to help propagate information among data sources: here WordNet, XWN, Wikipedia, FreeBase,…
- Different energy functions can be used for different types of relations, or a generic representation and generic relation symbols used for everything

# Open-Text Semantic Parsing (AISTATS 2012)

- Semantic Parsing: map a sentence into a Meaning Representation. Meaning Representation (MR): formal representation of the meaning. It can be in PROLOG, MySQL, ... or any structured language.
- Examples:
- • "What are the high points of states surrounding Mississippi ?"
  answer(A,(high point(B,A),state(B),next to(B,C),const(C,stateid(mississippi))))
- • "Show me flights from Boston to New York."
  SELECT flight id FROM flight WHERE from airport = 'boston' AND to airport = 'new york'
- Open-text: ability to handle any sentence regardless of its vocabulary (opposite to closed-domain).

# Processing Pipeline

- 3 steps:

``A musical score accompanies a television program ."

↓ ***Semantic Role Labeling***

(``A musical score", ``accompanies", ``a television program")

↓ ***Preprocessing (POS, Chunking, ...)***

((_musical_JJ score_NN ), _accompany_VB , _television_program_NN )

↓ ***Word-sense Disambiguation***

((_musical_JJ_1 score_NN_2), _accompany_VB_1, _television_program_NN_1)

- last formula defines the Meaning Representation (MR).

## Training Criterion

- Intuition: if an entity of a triplet was missing, we would like our model to predict it correctly i.e. to give it the lowest energy. For example, this would allow us to answer questions like "what is part of a car?"
- Hence, for any training triplet $x_i = (lhs_i, rel_i, rhs_i)$ we would like:

  (1) $E(lhs_i, rel_i, rhs_i) < E(lhs_j, rel_i, rhs_i)$,

  (2) $E(lhs_i, rel_i, rhs_i) < E(lhs_i, rel_j, rhs_i)$,

  (3) $E(lhs_i, rel_i, rhs_i) < E(lhs_i, rel_i, rhs_j)$,

That is, the energy function E is trained to rank training samples below all other triplets.

## Training Algorithm:

pseudo-likelihood + uniform sampling of negative variants

Train by stochastic gradient descent:

1. Randomly select a positive training triplet $x_i$ = ($lhs_i$, $rel_i$, $rhs_i$).
2. Randomly select constraint (1), (2) or (3) and an entity $\tilde{e}$:

- If constraint (1), construct negative triplet $\tilde{x}$ = ($\tilde{e}$, $rel_i$, $rhs_i$).
- Else if constraint (2), construct $\tilde{x}$ = ($lhs_i$, $\tilde{e}$, $rhs_i$).
- Else, construct $\tilde{x}$ = ($lhs_i$, $rel_i$, $\tilde{e}$).

3. If $E(x_i) > E(\tilde{x}) - 1$ make a gradient step to minimize:

   $\max(0, 1 - E(\tilde{x}) + E(x_i))$.

4. Constraint embedding vectors to norm 1

# Question Answering: implicitly adding new relations to WN

| | Model (All) | *TextRunner* |
|---|---|---|
| *lhs* | **_army_NN_1** | *army* |
| *rel* | **_attack_VB_1** | *attacked* |
| top ranked *rhs* | _troop_NN_4 | *Israel* |
| | _armed_service_NN_1 | *the village* |
| | _ship_NN_1 | *another army* |
| | _territory_NN_1 | *the city* |
| | _military_unit_NN_1 | *the fort* |
| top ranked *lhs* | _business_firm_NN_1 | *People* |
| | _person_NN_1 | *Players* |
| | _family_NN_1 | *one* |
| | _payoff_NN_3 | *Students* |
| | _card_game_NN_1 | *business* |
| *rel* | **_earn_VB_1** | *earn* |
| *rhs* | **_money_NN_1** | *money* |

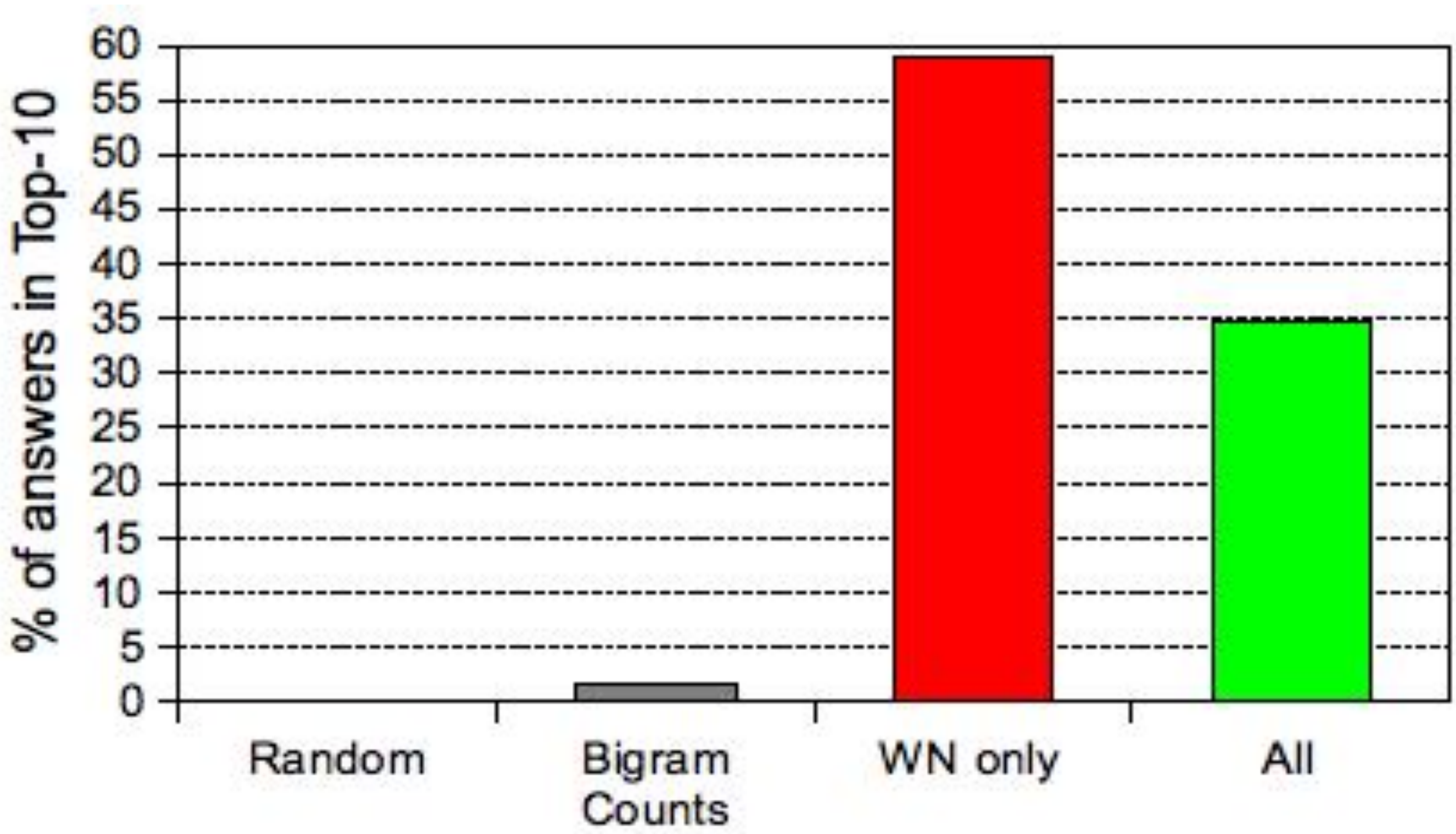
MRs inferred from text define triplets between WordNet synsets.

Model captures knowledge about relations between nouns and verbs.

→ Implicit addition of new relations to WordNet!

→ Generalize Freebase!

# Question Answering: Ranking Score



# Embedding Near Neighbors of Words & Senses

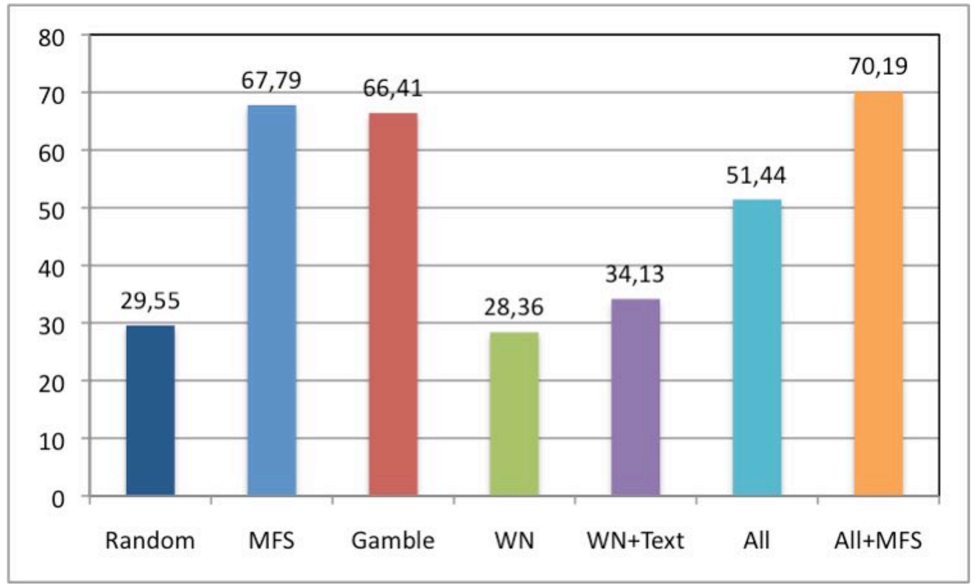
| _mark_NN | _mark_NN_1 | _mark_NN_2 |
|---|---|---|
| _indication_NN | _score_NN_1 | _marking_NN_1 |
| _print_NN_3 | _number_NN_2 | _symbolizing_NN_1 |
| _print_NN | _gradation_NN | _naming_NN_1 |
| _roll_NN | _evaluation_NN_1 | _marking_NN |
| _pointer_NN | _tier_NN_1 | _punctuation_NN_3 |

| _take_VB | _canary_NN | _different_JJ_1 |
|---|---|---|
| _bring_VB | _sea_mew_NN_1 | _eccentric_NN |
| _put_VB | _yellowbird_NN_2 | _dissimilar_JJ |
| _ask_VB | _canary_bird_NN_1 | _same_JJ_2 |
| _hold_VB | _larus_marinus_NN_1 | _similarity_NN_1 |
| _provide_VB | _mew_NN | _common_JJ_1 |

# Word Sense Disambiguation

- Senseval-3 results

(only sentences with
Subject-Verb-Object
structure)



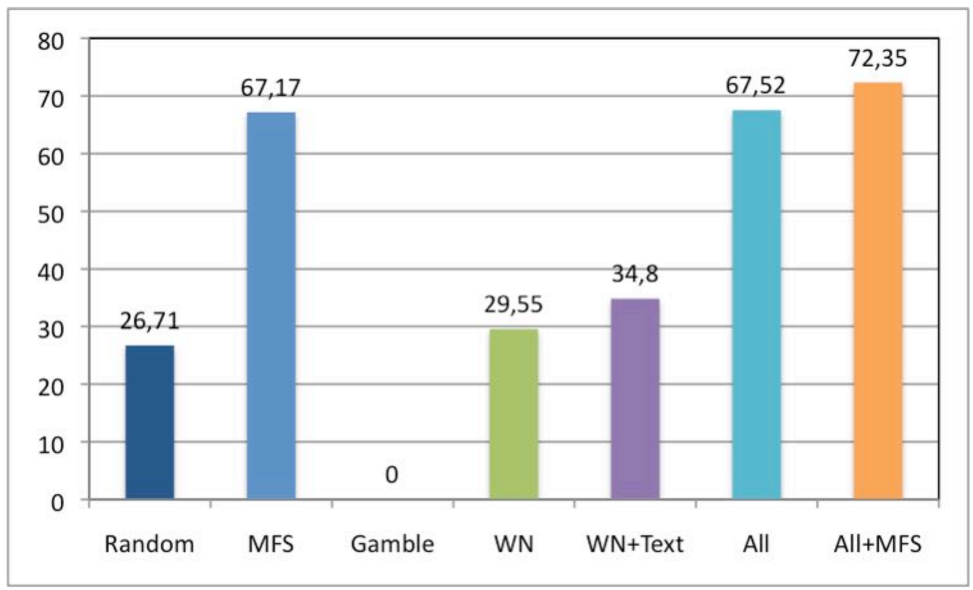

MFS=most frequent sense

All=training from all sources

Gamble=Decadt et al 2004
(Senseval-3 SOA)

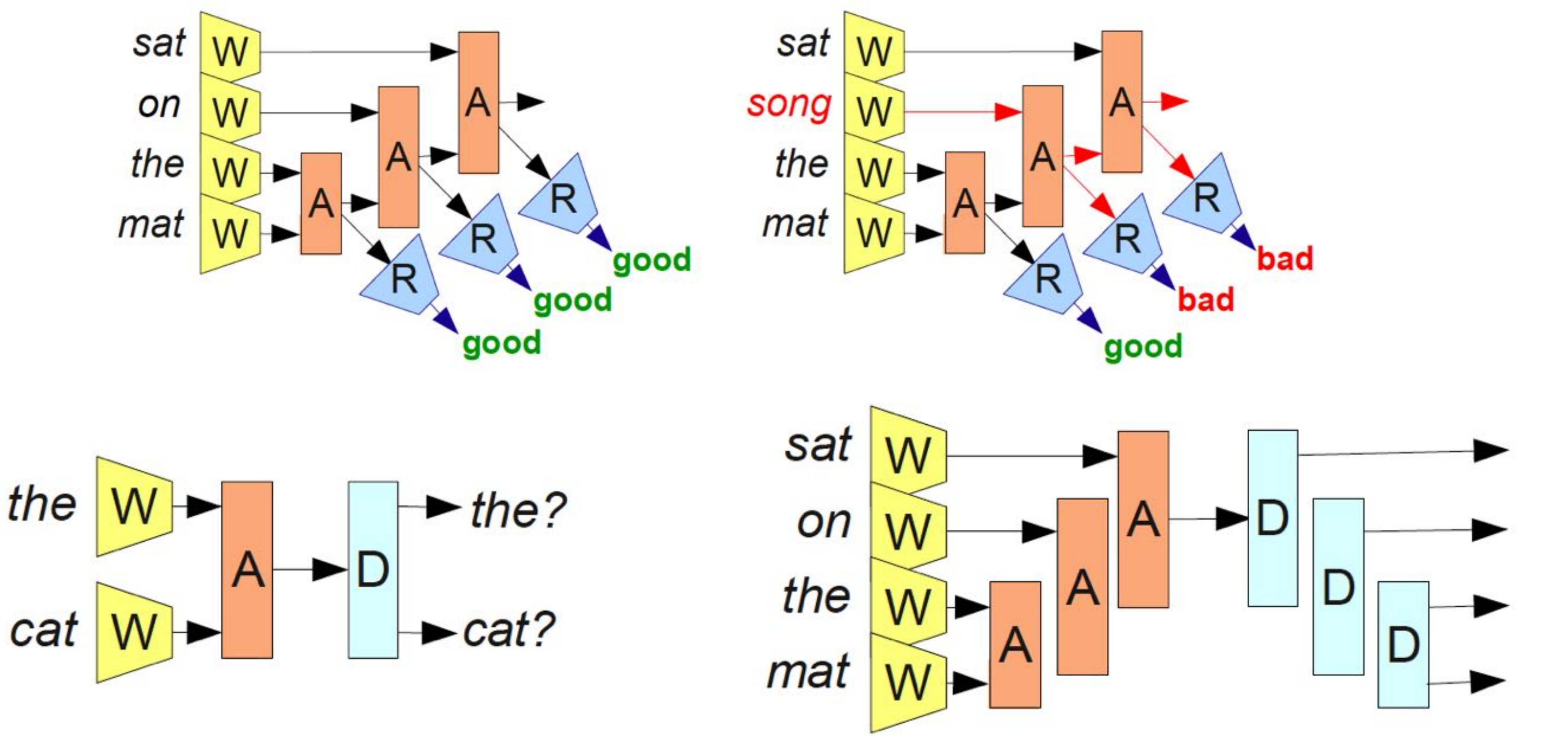
- XWN results

XWN = eXtended WN



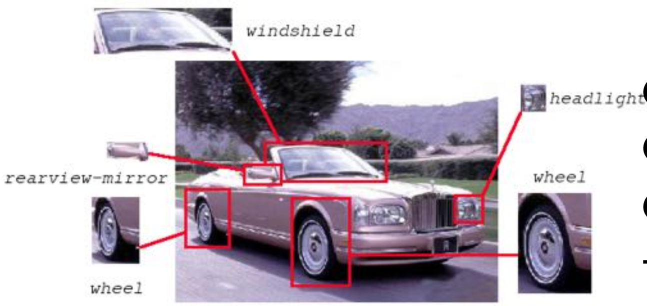
# Recursive Application of Relational Operators

Bottou 2011: ‘From machine learning to machine reasoning’, also Socher ICML2011.



# Relations on Multiple Data Types

- Add energy terms associated to relations from different data sources, shared embeddings



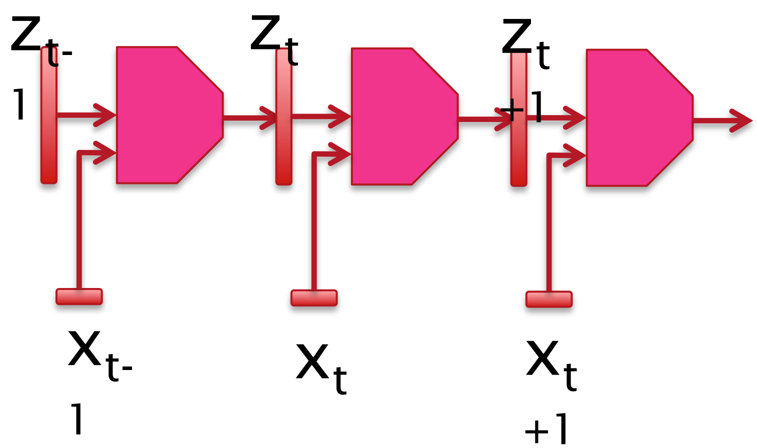

energy(object image, is-a, object label) +
energy(part image, is-a, part label) +
energy(part image, image-part-of, object image)
+ energy(part label, label-part-of, object label)

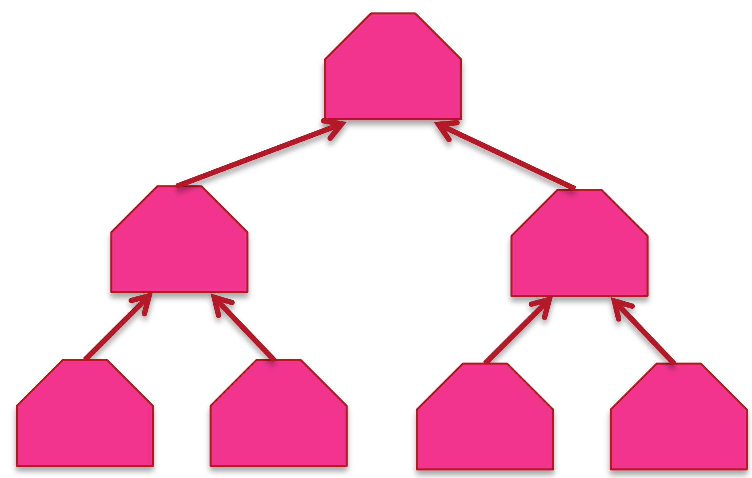
Table 1: **Summary of Test Set Results on ImageNet-WordNet.** Precision at 1 and 10, and Mean Average Precision (MAP) are given. (IW) resp. (I) refers to the (Image,Word) setup resp. (Image).

| | Image Annotation | | | Part-Object Detection | | | Triplet | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | p@1 | p@10 | MAP | p@1 | p@10 | MAP | p@1 | p@10 | MAP |
| Shared (IW) | 9.14% | 3.51% | 0.1768 | **11.48%** | **3.40%** | **0.1892** | 26.31% | **9.90%** | 0.5545 |
| UnShared (IW) | 9.45% | 3.68% | 0.1847 | 10.01% | 3.02% | 0.1669 | **33.13%** | 9.62% | **0.5595** |
| Shared (I) | 11.21% | 3.85% | 0.2021 | 5.13% | 1.84 % | 0.0955 | 11.21% | 3.85% | 0.2021 |
| UnShared (I) | **12.94%** | **4.10%** | **0.2219** | 6.08% | 2.11% | 0.1118 | 12.94% | 4.10% | 0.2219 |
| SVM | 10.02% | 3.72% | 0.1864 | – | – | – | 10.02% | 3.72% | 0.1864 |

## Recurrent and Recursive Nets

- Replicate a parametrized function over different time steps or nodes of a DAG
- Output state at one time-step / node is used as input for another time-step / node
- Very deep once unfolded!



# Conclusion

- AI → learning → representation-learning
- Deep learning to disentangle factors of variation and discover representations for higher-level abstractions
- No immediate generalization from discrete spaces → learn a distributed semantic representation for discrete objects
- Word embeddings generalize across semantically similar words
- Combine word embeddings into representations and energy functions for phrases and relations
- Applications to language modeling (speech recognition, language translation), sentiment analysis, parsing, paraphrasing, word sense disambiguation, question answering…

`LISA team:` **Merci! Questions?**