Max-margin learning

Nando de Freitas

UNIVERSITY OF OXFORD

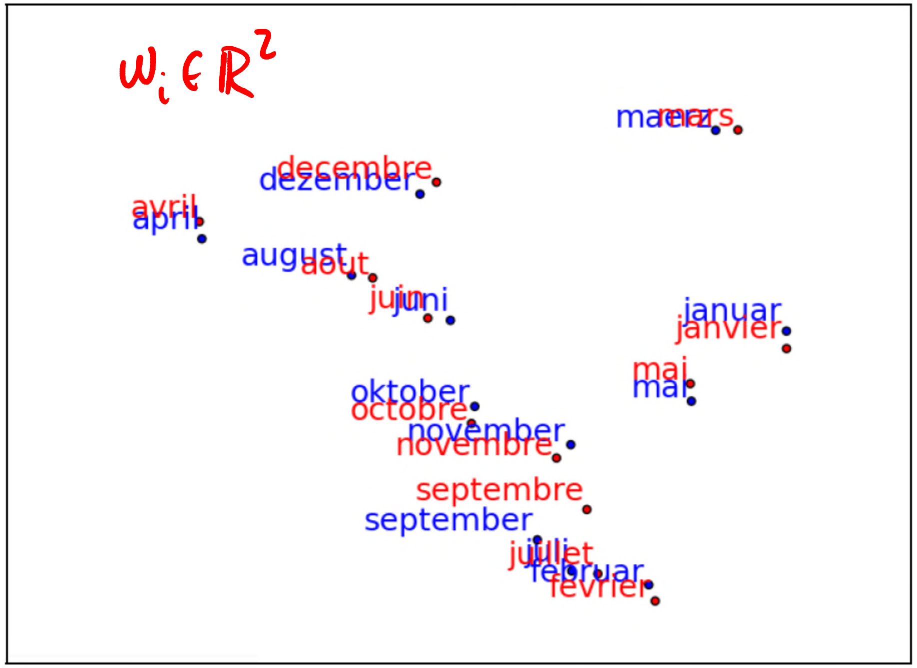
Outline of the lecture

Max margin learning is an extremely powerful idea for learning features with auxiliary tasks, and then use these features to solve tasks with few data. The goal of this lecture is for you to learn

- ❑ Transfer, multi-task, and multi-instance learning
- ❑ Harnessing auxiliary tasks to learn features
    - ❑ Matchings
    - ❑ Preferences
    - ❑ Corruption
- ❑ Formulations of multi-task learning
- ❑ Applications:
    - ❑ Cross-lingual embeddings
    - ❑ Relation learning
    - ❑ Question answering
    - ❑ Memory networks

Embedding discrete objects in metric spaces

Code
Words
Formulae
Logical expressions
Symbols
DNA sequences

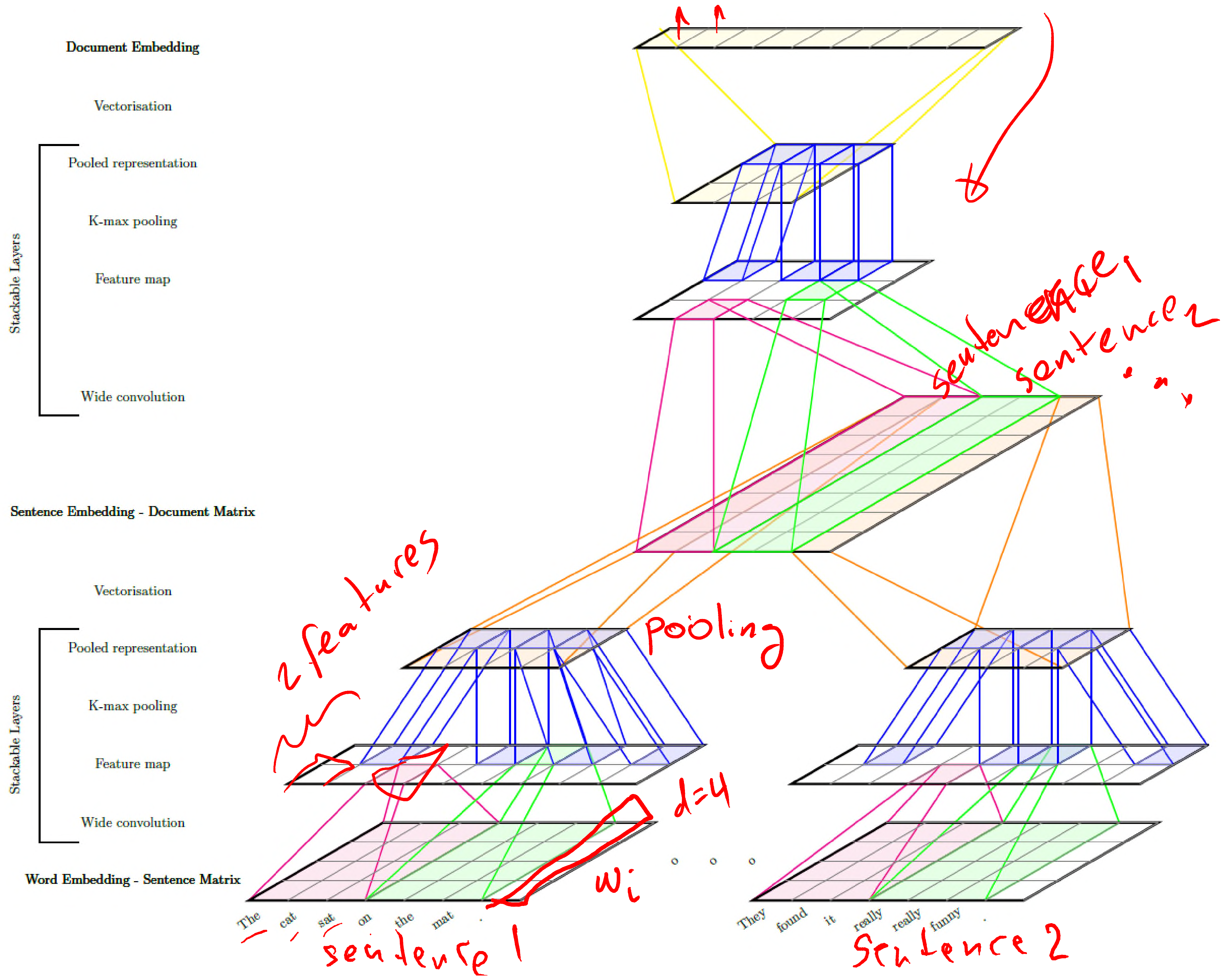


$w_i \in \mathbb{R}^d$

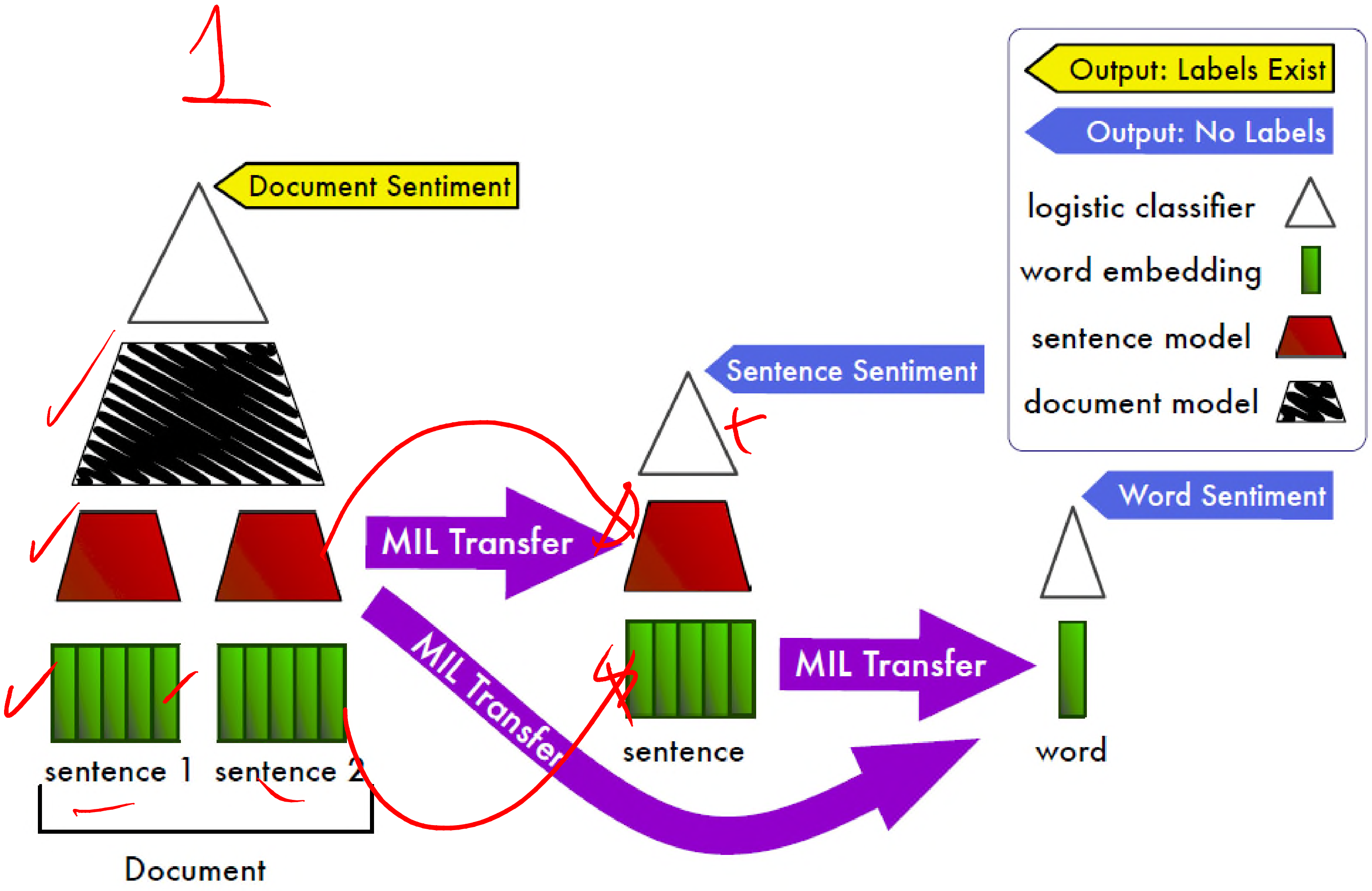
$$\begin{array}{c} 1 \\ \\ d \end{array}\begin{bmatrix} w_3 \\ \\ \end{bmatrix} = \underset{d\times K}{\begin{bmatrix} w_1 & w_2 & \boxed{w_3} & \cdots & w_K \end{bmatrix}} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\begin{array}{c} 1 \\ \\ \\ \\ \\ K \end{array}$$

d-dimensional encoding of word.

vector of zeros with a 1 indicating a word token in the vocabulary.


Document Embedding
Vectorisation
Stackable Layers
Pooled representation
K-max pooling
Feature map
Wide convolution
Sentence Embedding - Document Matrix
Vectorisation
Stackable Layers
Pooled representation
K-max pooling
Feature map
Wide convolution
Word Embedding - Sentence Matrix
The cat sat on the mat .
They found it really really funny .
o o o
sentence 1
sentence 2
. . .
2 features
Pooling
d=4
w_i
sentence 1
Sentence 2

Idea: learn embeddings (features) in one task and transfer these to solve new tasks



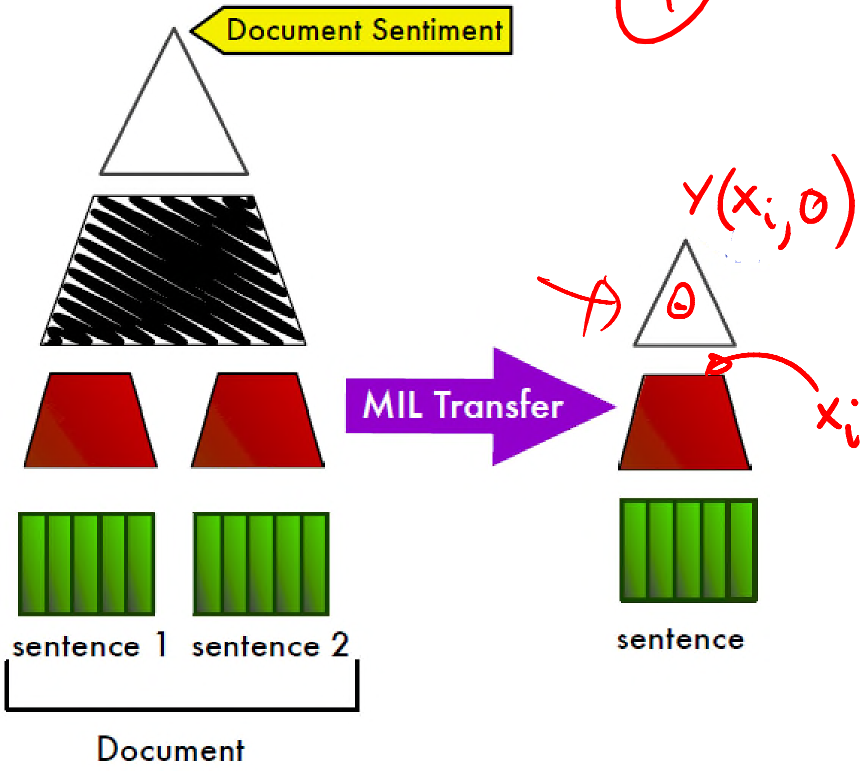
[Kotzias, Denil & NdF, 2014]

Deep Multi-Instance Learning

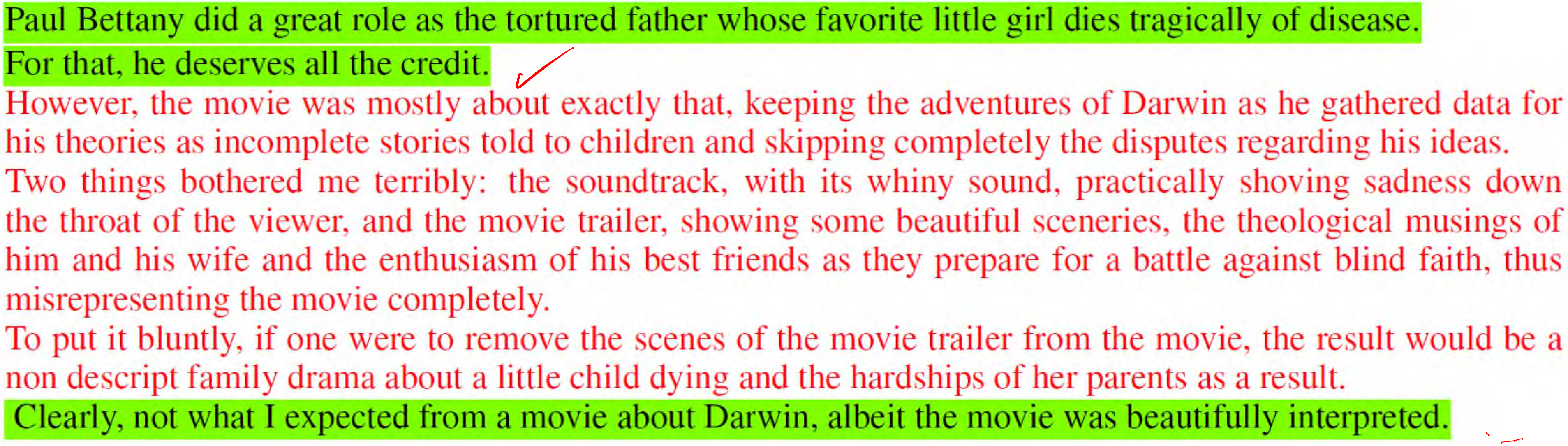
$$\underline{W(\mathbf{x}_i,\mathbf{x}_j)} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2) \leftarrow \text{similarity}$$

sentence embeddings

$$J(\boldsymbol{\theta}) = \sum_{i,j\in I} \underline{W(\mathbf{x}_i,\mathbf{x}_j)}\,\underbrace{(\underline{y(\mathbf{x}_i,\boldsymbol{\theta})} - \underline{y(\mathbf{x}_j,\boldsymbol{\theta})})^2}_{①} + \underbrace{\underline{\lambda}\sum_{g\in G}\left(\underbrace{\frac{1}{|g|}\sum_{i\in g} y(\mathbf{x}_i,\boldsymbol{\theta})} - \underline{s_g}\right)^2}_{②}$$



Deep Multi-Instance Learning



Paul Bettany did a great role as the tortured father whose favorite little girl dies tragically of disease.

For that, he deserves all the credit.

However, the movie was mostly about exactly that, keeping the adventures of Darwin as he gathered data for his theories as incomplete stories told to children and skipping completely the disputes regarding his ideas.

Two things bothered me terribly: the soundtrack, with its whiny sound, practically shoving sadness down the throat of the viewer, and the movie trailer, showing some beautiful sceneries, the theological musings of him and his wife and the enthusiasm of his best friends as they prepare for a battle against blind faith, thus misrepresenting the movie completely.

To put it bluntly, if one were to remove the scenes of the movie trailer from the movie, the result would be a non descript family drama about a little child dying and the hardships of her parents as a result.

Clearly, not what I expected from a movie about Darwin, albeit the movie was beautifully interpreted.

Auxiliary tasks to learn features that can be transferred to learn tasks with few labels

1. Matchings

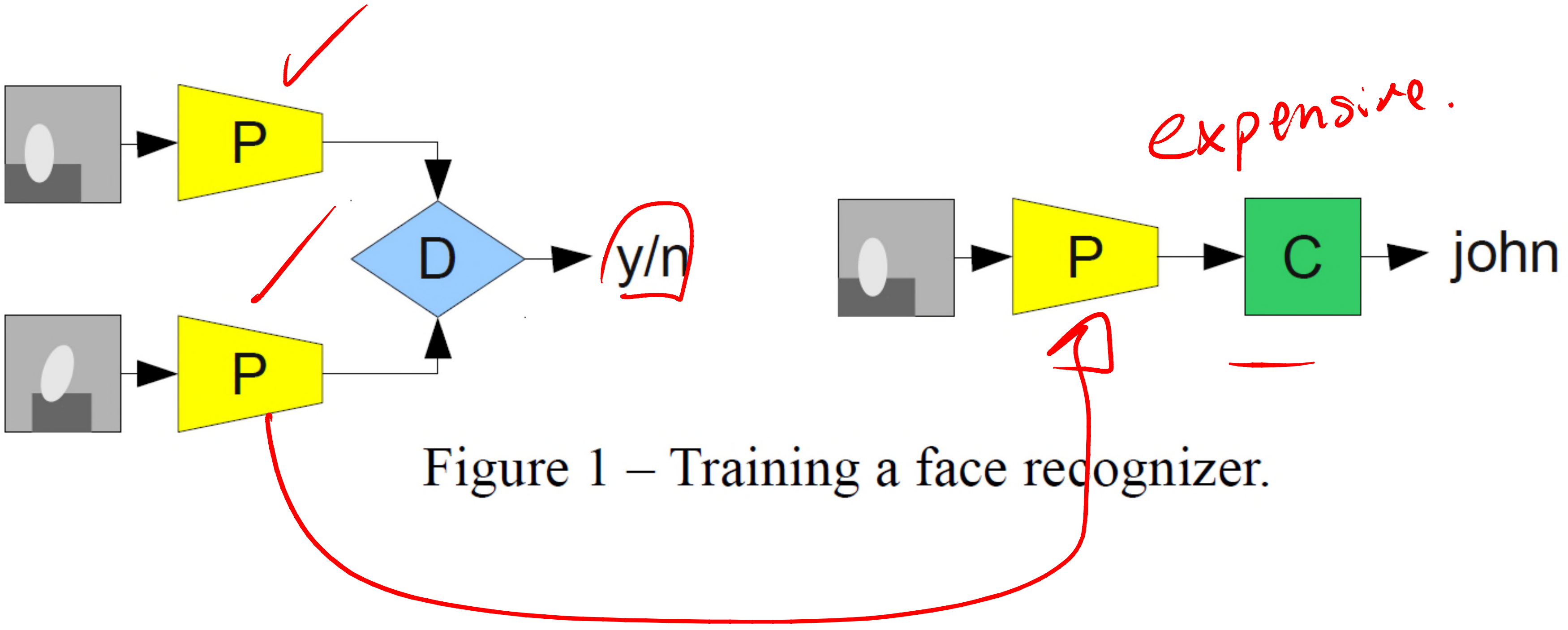
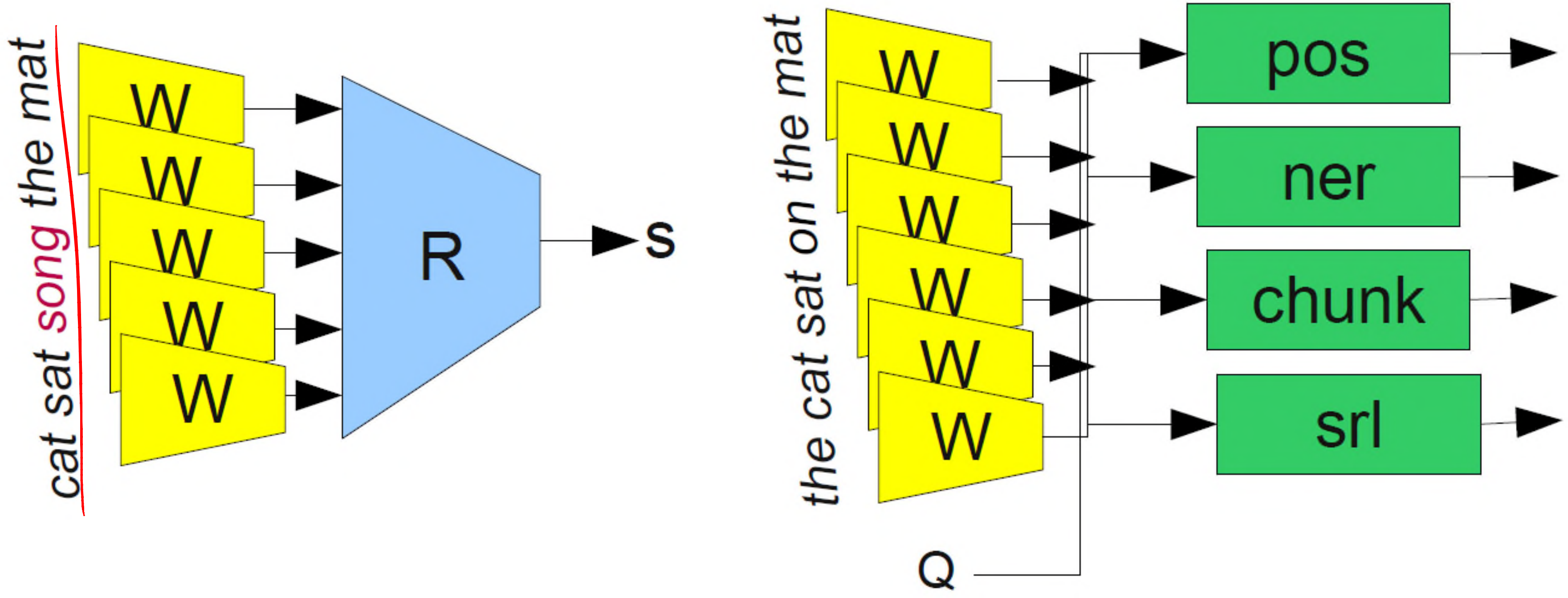


Figure 1 – Training a face recognizer.

[*From machine learning to machine reasoning*, Leon Bottou]

Auxiliary tasks to learn features that can be transferred to learn tasks with few labels

2. Corruption



[*NLP almost from scratch*, Ronan Collobert and colleagues]

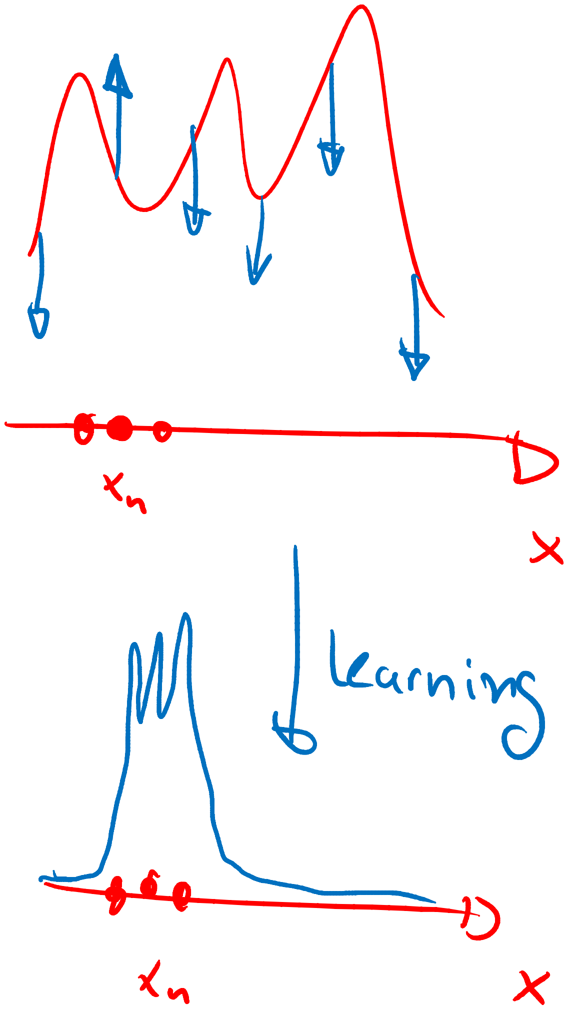
Auxiliary tasks to learn features that can be transferred to learn tasks with few labels

1. Preferences

Max-margin formulations

data instance $\mathbf{x}_n \in \mathcal{X}$

$$
\begin{aligned}
p(\mathbf{x}_n|\boldsymbol{\theta}) &= \frac{1}{Z(\underline{\boldsymbol{\theta}})} \exp[-\underline{E(\mathbf{x}_n,\boldsymbol{\theta})}] \\
Z(\boldsymbol{\theta}) &= \int_{\mathcal{X}} \exp[-E(\mathbf{x}_n,\boldsymbol{\theta})]\,d\mathbf{x}_n
\end{aligned}
$$



$$
\boldsymbol{\theta}^\star = \mathop{\arg\min}_{\boldsymbol{\theta}:\ p(\underline{\mathbf{x}_n}|\boldsymbol{\theta}) \geq p(\underline{\mathbf{x}}|\boldsymbol{\theta}) + m_p} \underline{\|\boldsymbol{\theta}\|_2^2}
$$

$$
\forall n,\ \forall \mathbf{x} : \|\mathbf{x} - \mathbf{x}_n\| > \epsilon
$$

Max-margin formulations

$$\boldsymbol{\theta}^\star = \mathop{\arg\min}_{\boldsymbol{\theta}: E(\underline{\mathbf{x}},\boldsymbol{\theta}) \geq \underline{E(\mathbf{x}_n,\boldsymbol{\theta})} + m} \|\boldsymbol{\theta}\|_2^2$$

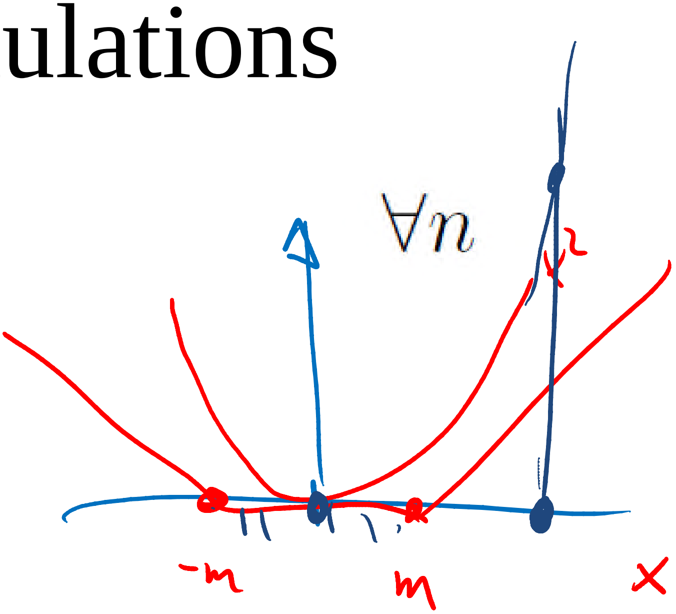
$$\forall n,\ \forall \mathbf{x} : \|\underline{\mathbf{x}} - \mathbf{x}_n\| > \underline{\epsilon}$$

[*A tutorial on energy based learning*, Yann LeCun et al]

Max-margin formulations

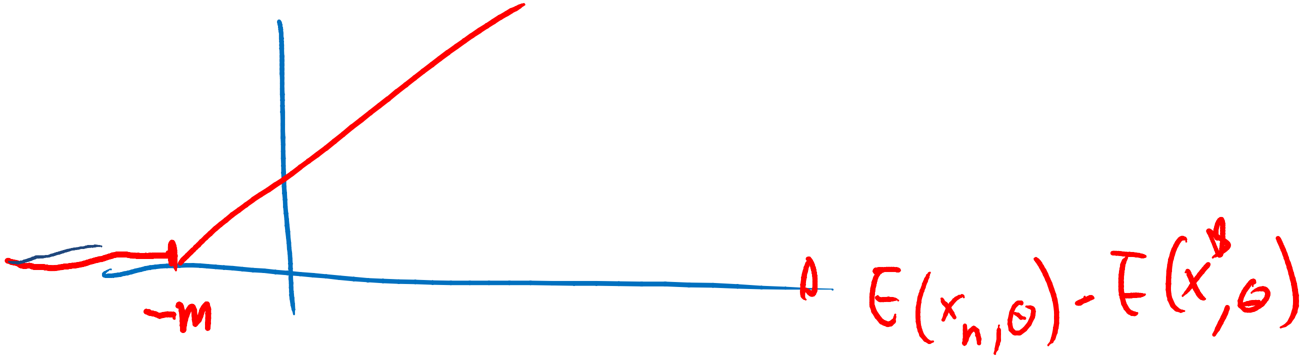
$$\theta^\star = \mathop{\arg\min}_{\theta : E(\mathbf{x}^\star, \theta) \geq E(\mathbf{x}_n, \theta) + m} ||\theta||_2^2 \qquad \forall n$$

$$\mathbf{x}^\star = \mathop{\arg\min}_{\mathbf{x} : ||\mathbf{x} - \mathbf{x}_n|| > \epsilon} E(\mathbf{x}, \theta)$$



Hinge Loss unconstrained formulation:

$$\theta^\star = \mathop{\arg\min}_{\theta} \left\{ \sum_{n=1}^{N} \max\left[0, \underbrace{m + E(\mathbf{x}_n, \theta) - E(\mathbf{x}^\star, \theta)}_{}\right] + \lambda ||\theta||_2^2 \right\}$$



Max-margin formulations

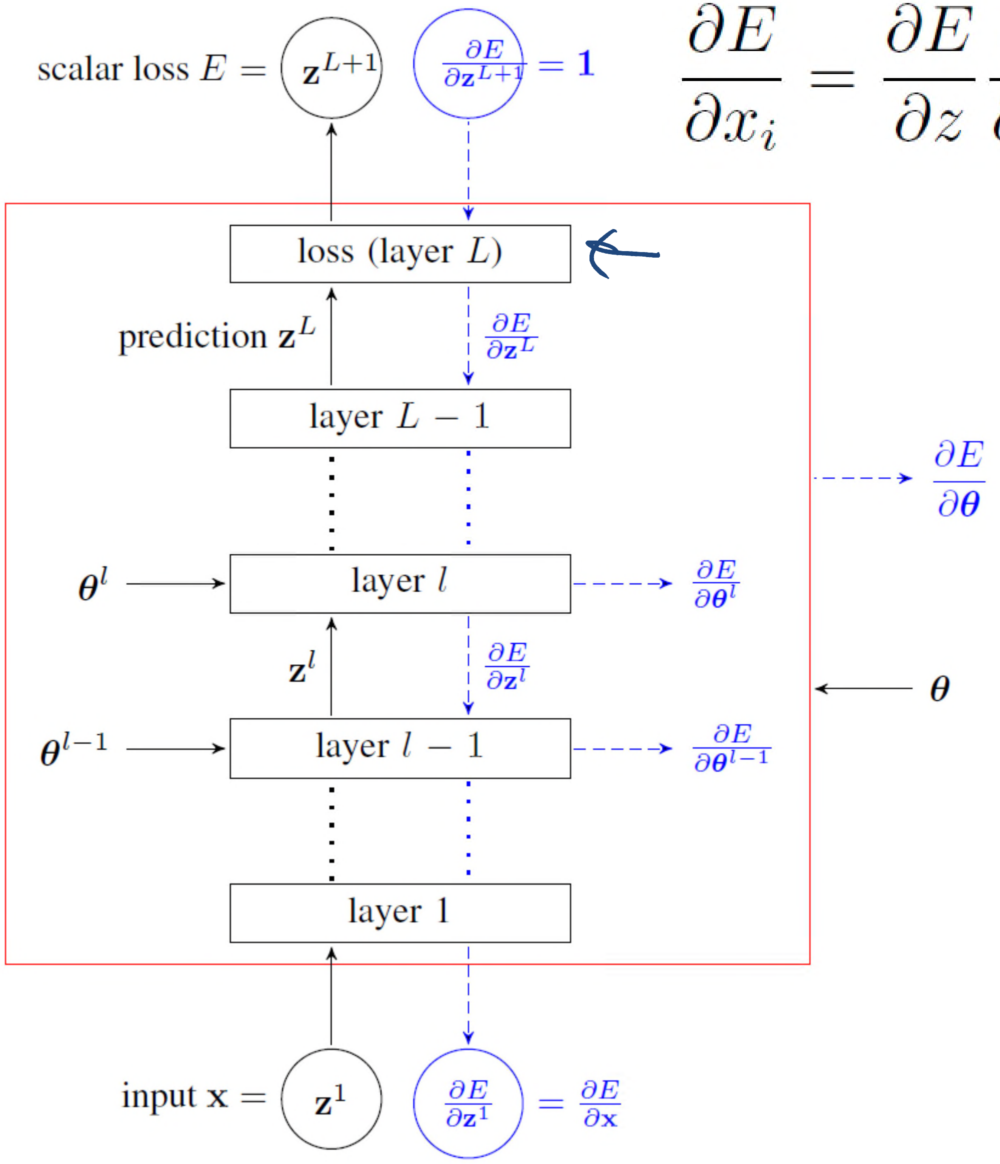
introduce corrupted data $\widetilde{\mathbf{x}}_n$

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \left\{ \sum_{n=1}^{N} \max\left[0, m_e + \underline{E(\overset{\text{correct}}{\mathbf{x}_n}, \boldsymbol{\theta})} - \overset{\text{incorrect}}{E(\widetilde{\mathbf{x}}_n, \boldsymbol{\theta})}\right] + \lambda \|\boldsymbol{\theta}\|_2^2 \right\}$$

Hinge loss layer

$$z = E(\underline{x_1}, \underline{x_2}, m) = \max\left[0, m + \overset{+1}{\underline{x_2}} - \overset{-1}{\underline{x_1}}\right]$$

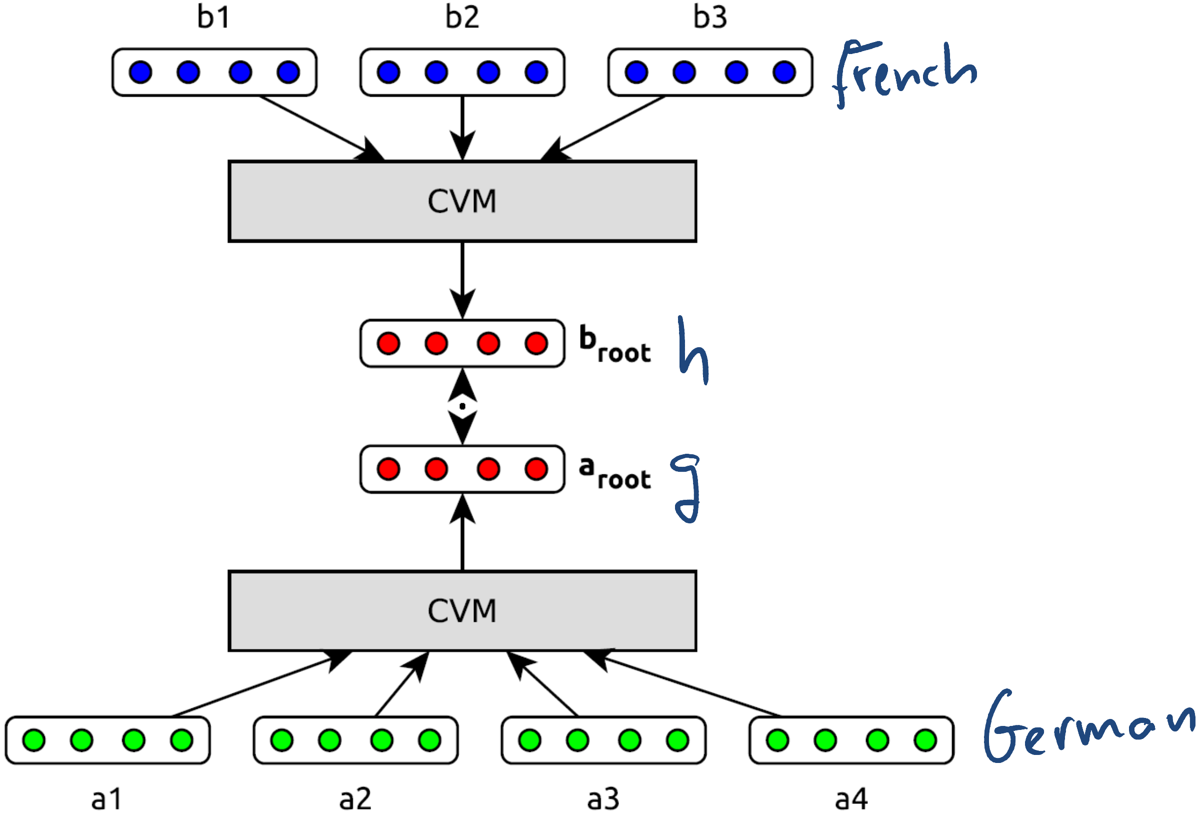
$$\frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial z}\frac{\partial z}{\partial x_i} = (-1)^{i}\mathbb{I}_{(m+x_2-x_1>0)}$$


scalar loss $E = \mathbf{z}^{L+1}$
$\frac{\partial E}{\partial \mathbf{z}^{L+1}} = \mathbf{1}$
loss (layer L)
$\leftarrow$
prediction $\mathbf{z}^L$
$\frac{\partial E}{\partial \mathbf{z}^L}$
layer $L-1$
$\frac{\partial E}{\partial \boldsymbol{\theta}}$
$\boldsymbol{\theta}^l$
layer l
$\frac{\partial E}{\partial \boldsymbol{\theta}^l}$
$\mathbf{z}^l$
$\frac{\partial E}{\partial \mathbf{z}^l}$
$\boldsymbol{\theta}$
$\boldsymbol{\theta}^{l-1}$
layer $l-1$
$\frac{\partial E}{\partial \boldsymbol{\theta}^{l-1}}$
layer 1
input $\mathbf{x} = \mathbf{z}^1$
$\frac{\partial E}{\partial \mathbf{z}^1} = \frac{\partial E}{\partial \mathbf{x}}$

Example: Bi-lingual word embeddings

$$E_{bi}(\mathbf{a},\mathbf{b},\boldsymbol{\theta}) = \|\mathbf{g}(\underline{\mathbf{a},\boldsymbol{\theta}_g}) - \mathbf{h}(\underline{\mathbf{b},\boldsymbol{\theta}_h})\|^2 \checkmark$$

$$L_{hinge}(\mathbf{a},\mathbf{b},\mathbf{n},\boldsymbol{\theta}) = \max\left[0, \underline{m} + E_{bi}(\mathbf{a},\mathbf{b},\boldsymbol{\theta}) - \underline{E_{bi}}(\mathbf{a},\underset{\substack{\uparrow \\ \text{Corruption}}}{\mathbf{n}},\boldsymbol{\theta})\right]$$



[Karl Hermann and Phil Blunsom, 2014]

Siamese networks (Yann LeCun)

$$L_{siamese}(\boldsymbol{\theta}, S_{ij}) = \begin{cases} ||\mathbf{h}(\mathbf{x}_i;\boldsymbol{\theta}) - \mathbf{h}(\mathbf{x}_j;\boldsymbol{\theta})||_2^2 & \text{if } S_{ij} = 1 \\ \max\left[0, \underline{m_h} - ||\mathbf{h}(\mathbf{x}_i;\boldsymbol{\theta}) - \mathbf{h}(\mathbf{x}_j;\boldsymbol{\theta})||_2^2\right] & \underline{\text{if } S_{ij} = 0} \end{cases}$$

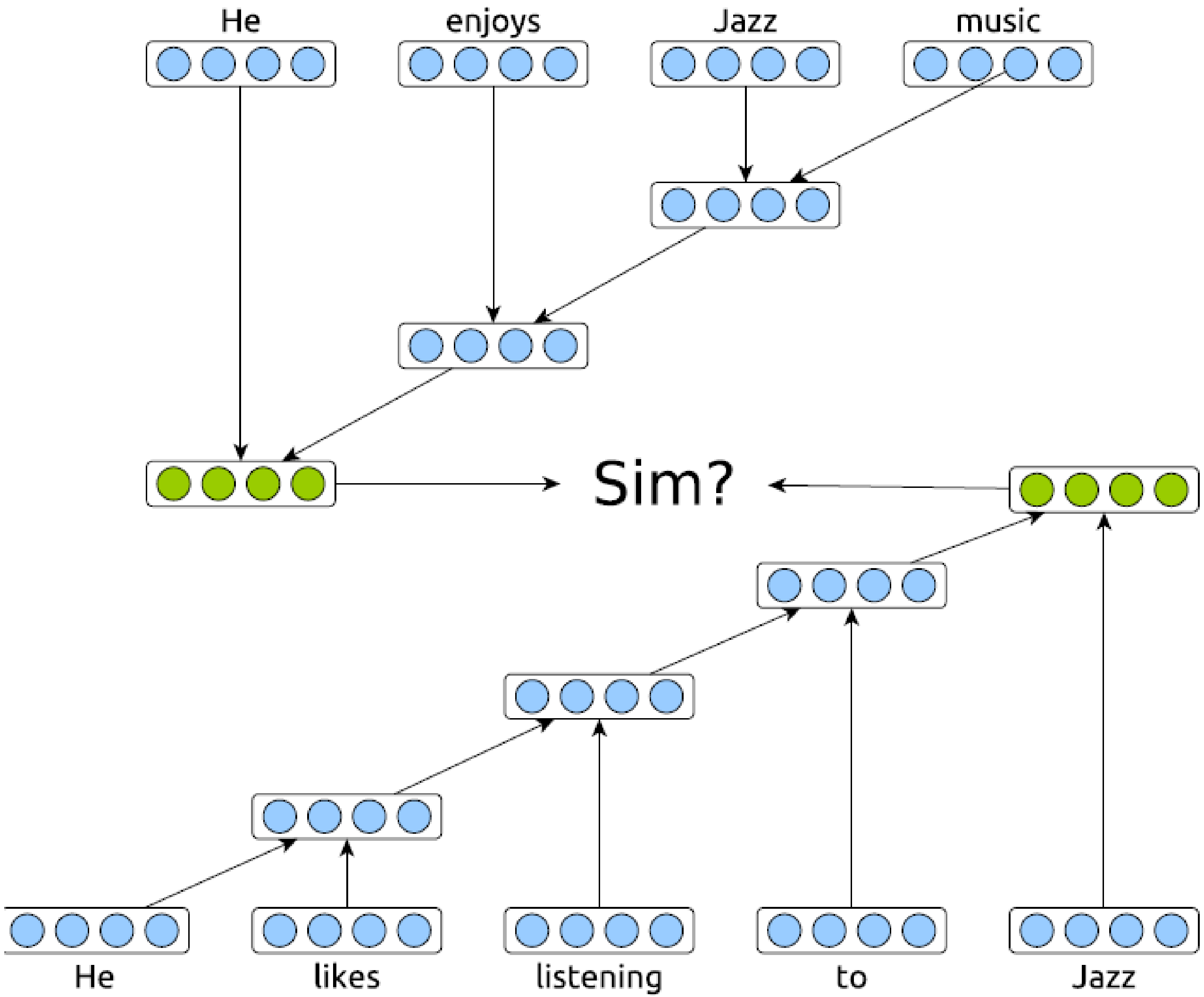
Semi-supervised deep learning (Jason Weston et al)

$$\sum_{n \in \mathcal{L}} \underline{\text{NLL}(\mathbf{h}(\mathbf{x}_n;\boldsymbol{\theta}), \mathbf{y}_n)} + \lambda \sum_{i,j \in \mathcal{U}} L_{siamese}(\boldsymbol{\theta}, S_{ij})$$

↑ labels

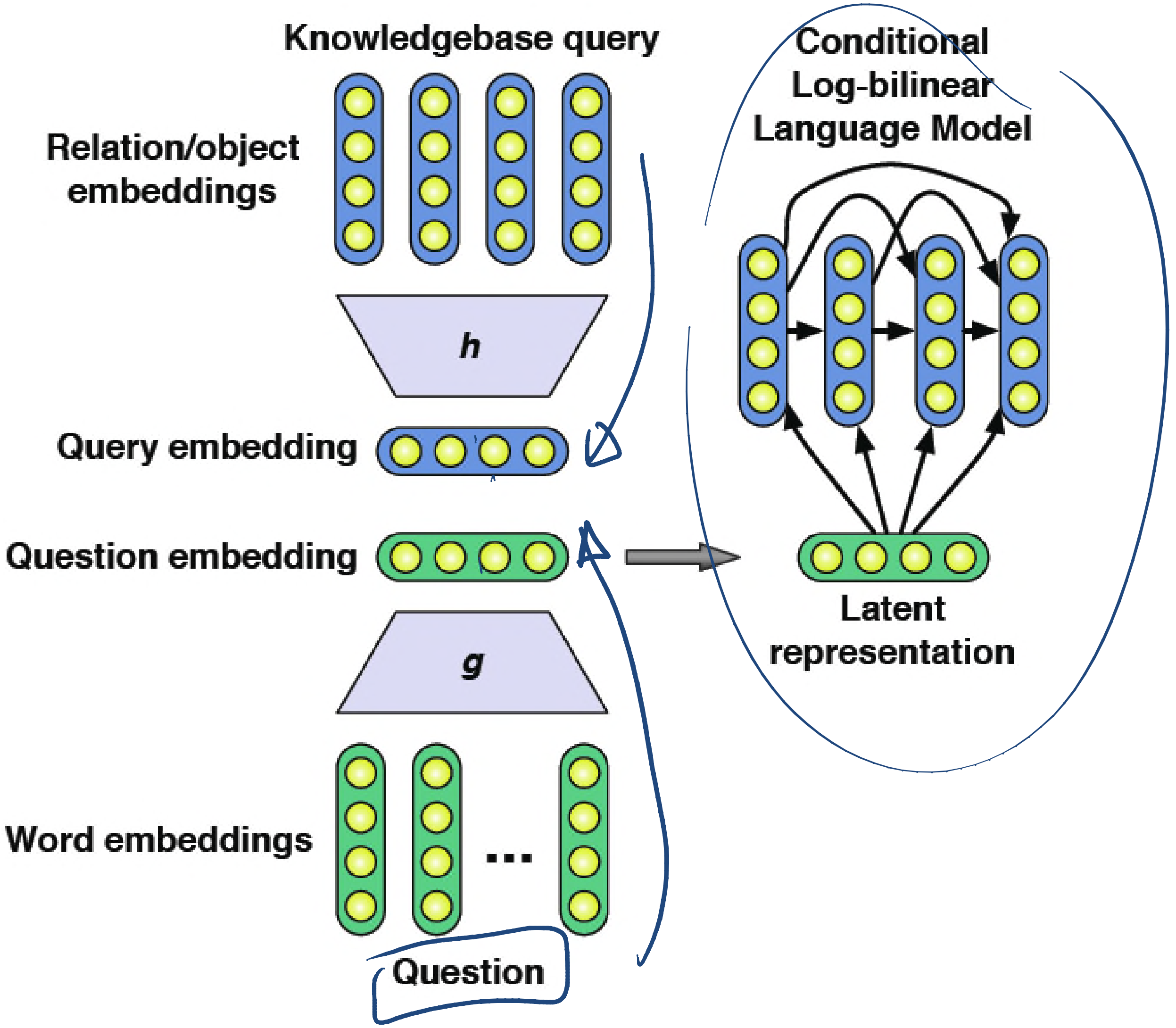
no labels.

Applications: Paraphrase detection



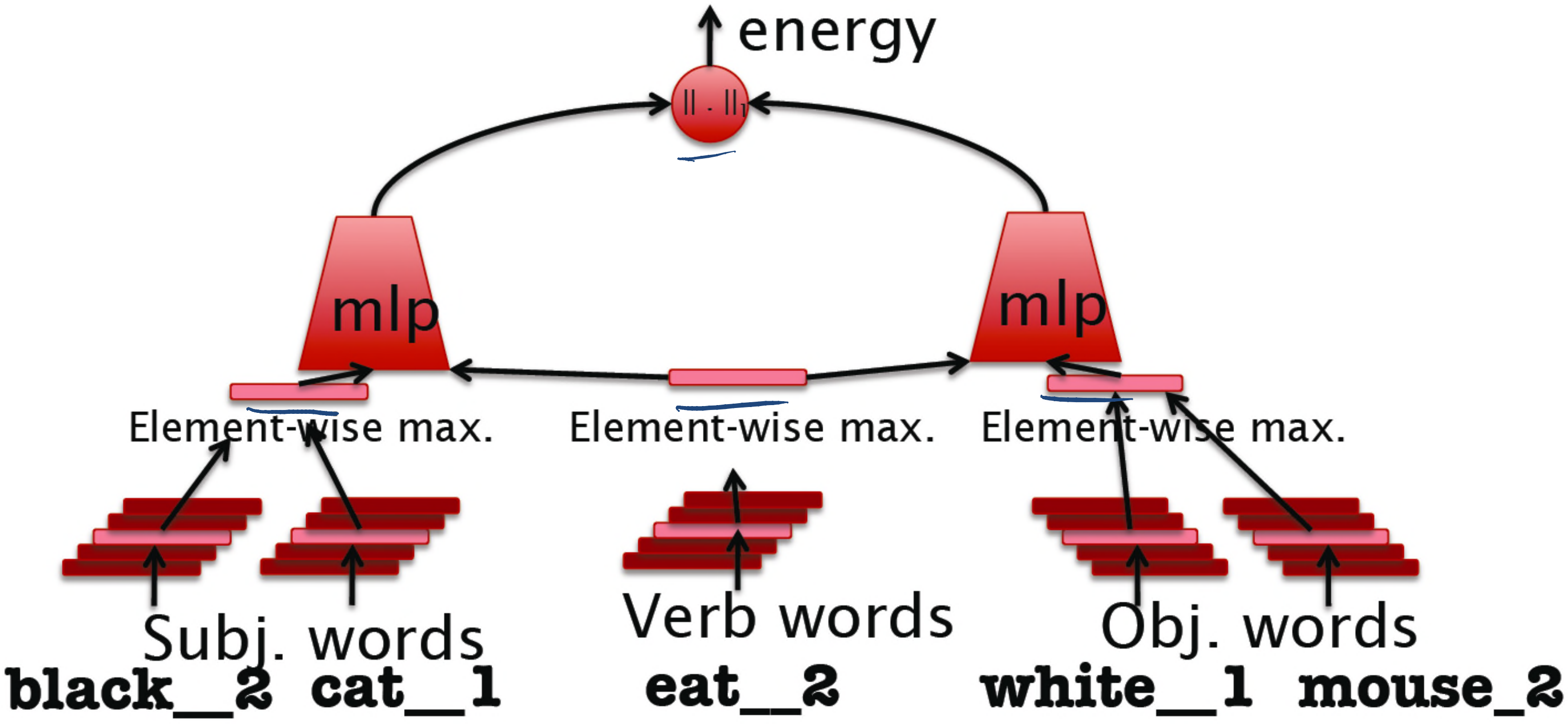
[Phil Blunsom et al]

Transfer: Question answering



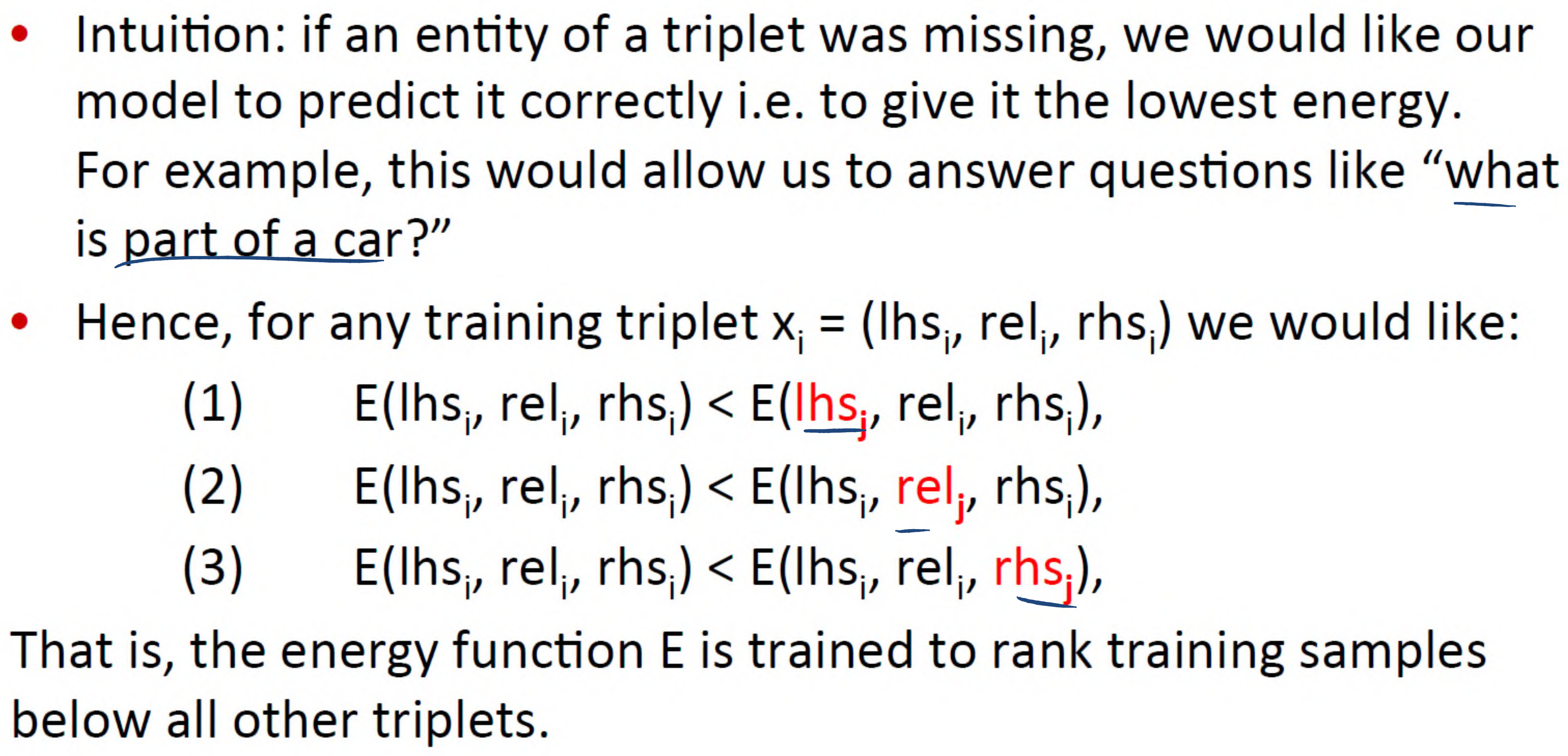
[Grefenstette, Blunsom, NdF, Hermann, 2014]

Relation learning



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

Relation learning



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

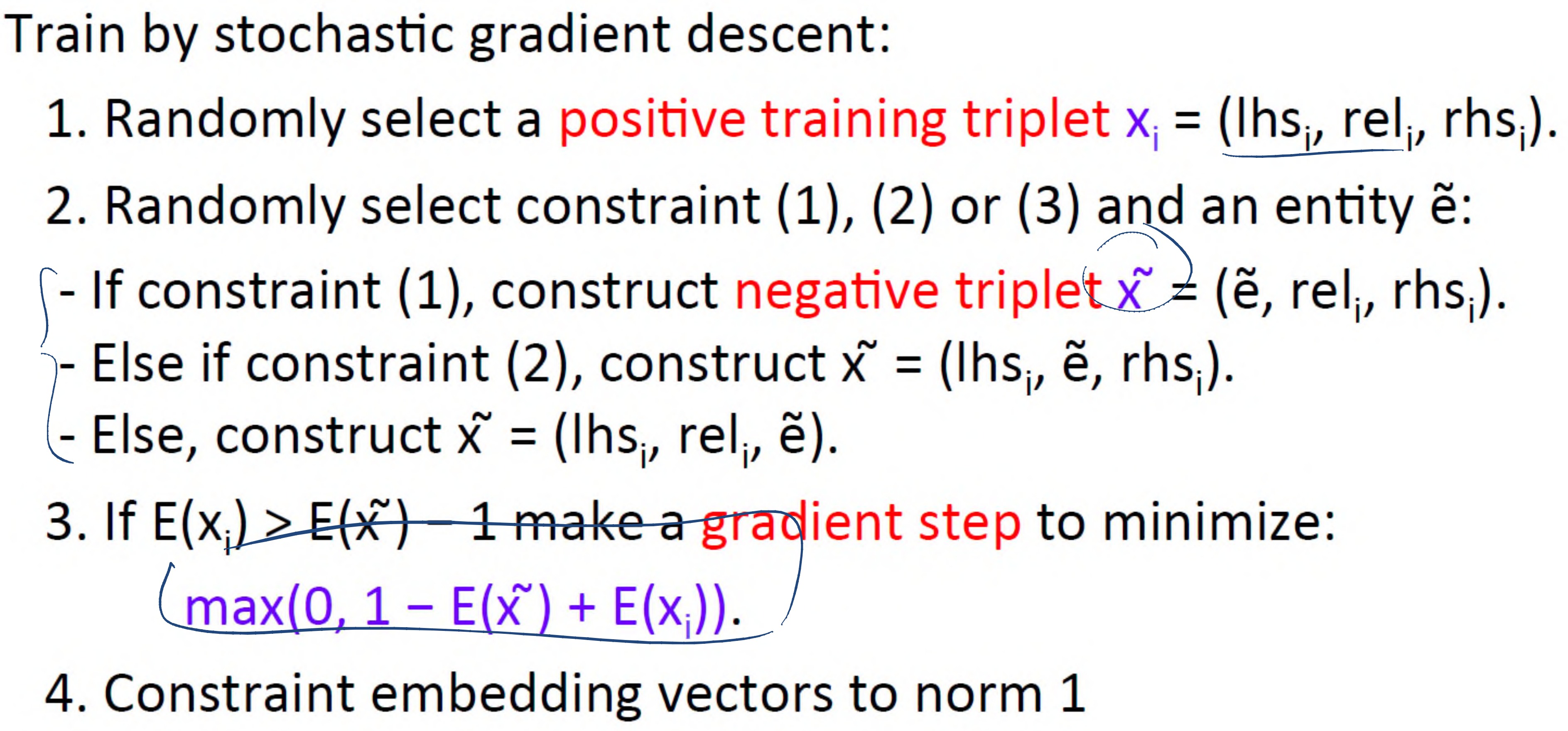
(2) $E(lhs_i, rel_i, rhs_i) < E(lhs_i, rel_j, rhs_i)$,

(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.

[Yoshua Bengio et al]

Relation learning



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 $\tilde{x} = (lhs_i, rel_i, \tilde{e})$.

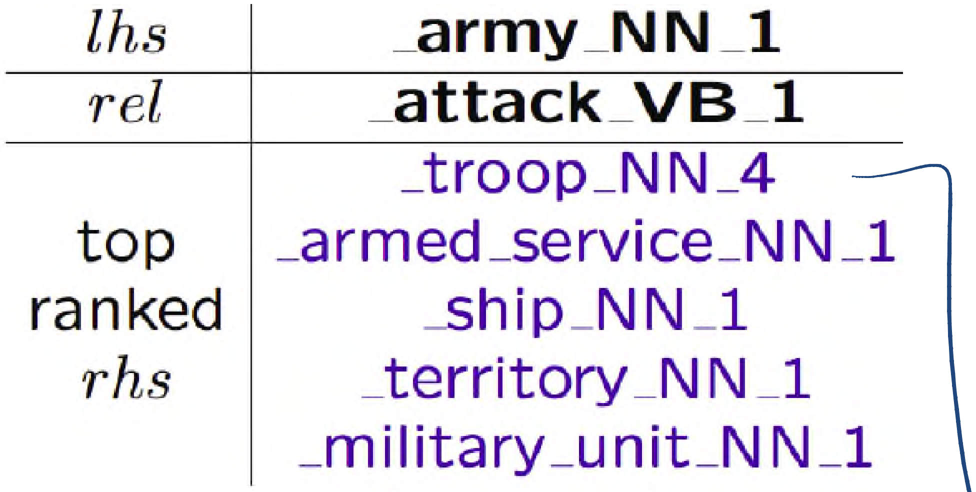
3. If $E(x_i) > E(\tilde{x}) - 1$ make a gradient step to minimize:

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

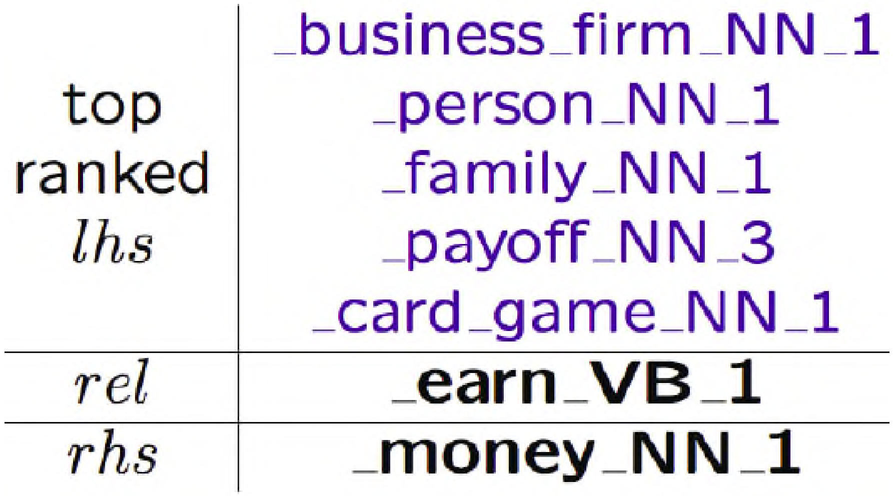
4. Constraint embedding vectors to norm 1

[Yoshua Bengio et al]

Relation learning



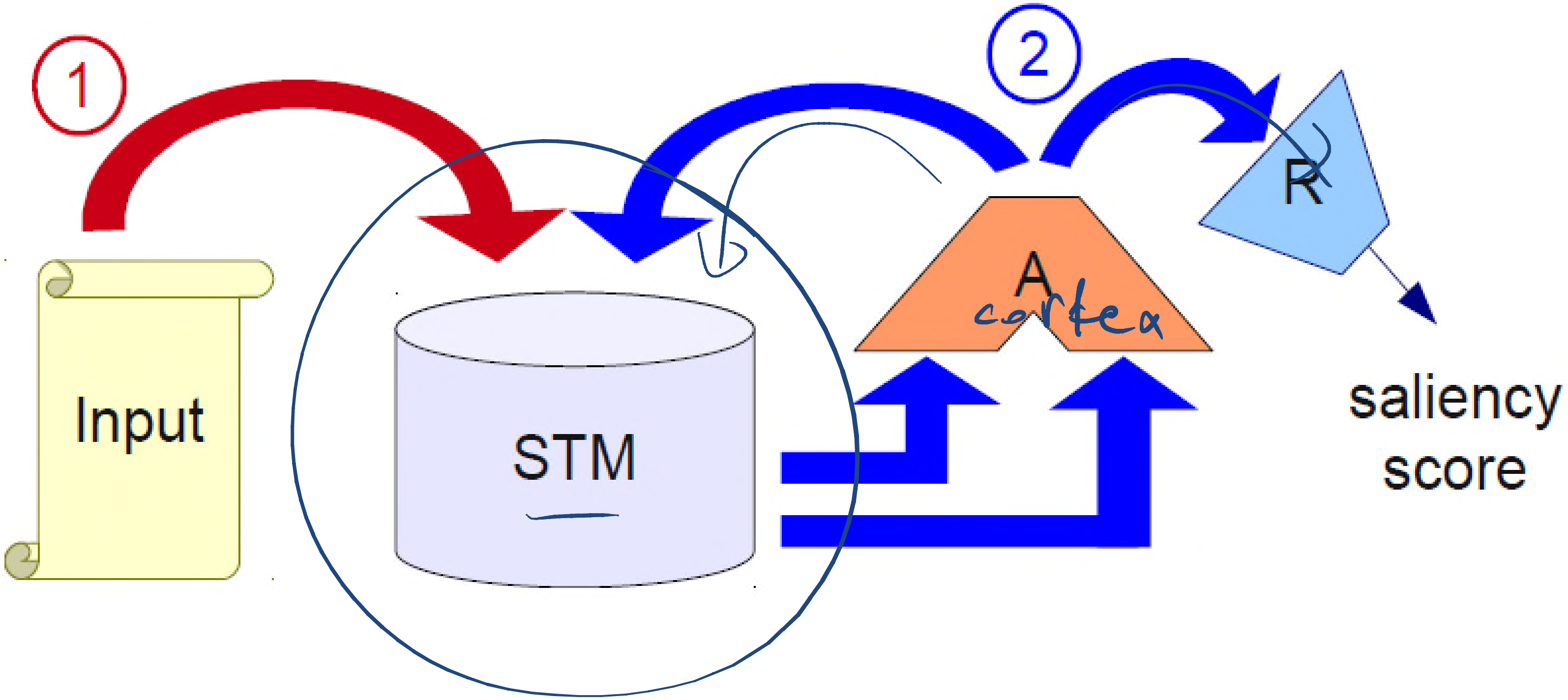
lhs	**_army_NN_1**
rel	**_attack_VB_1**
top ranked *rhs*	_troop_NN_4 _armed_service_NN_1 _ship_NN_1 _territory_NN_1 _military_unit_NN_1



top ranked *lhs*	_business_firm_NN_1 _person_NN_1 _family_NN_1 _payoff_NN_3 _card_game_NN_1
rel	**_earn_VB_1**
rhs	**_money_NN_1**

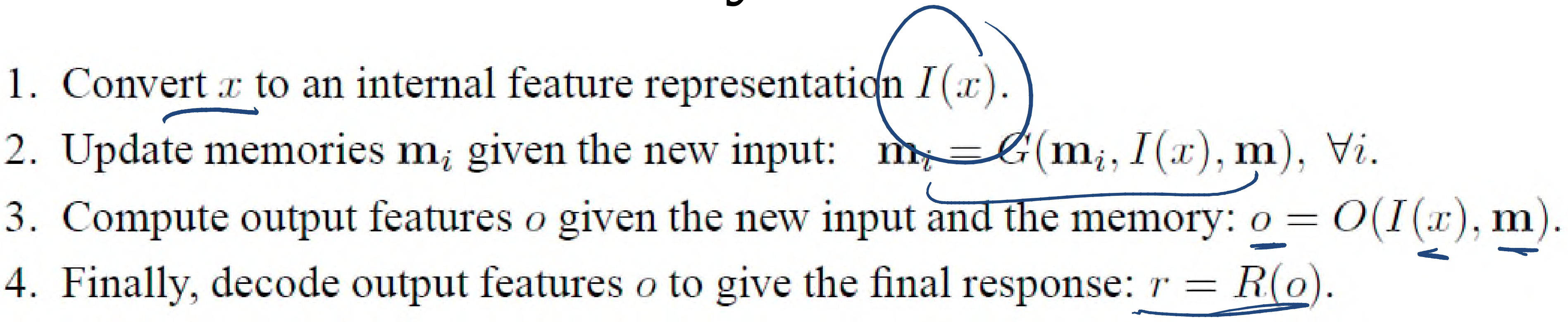
[Yoshua Bengio et al]

Short term & long term memory



[*From machine learning to machine reasoning*, Leon Bottou]

Memory networks



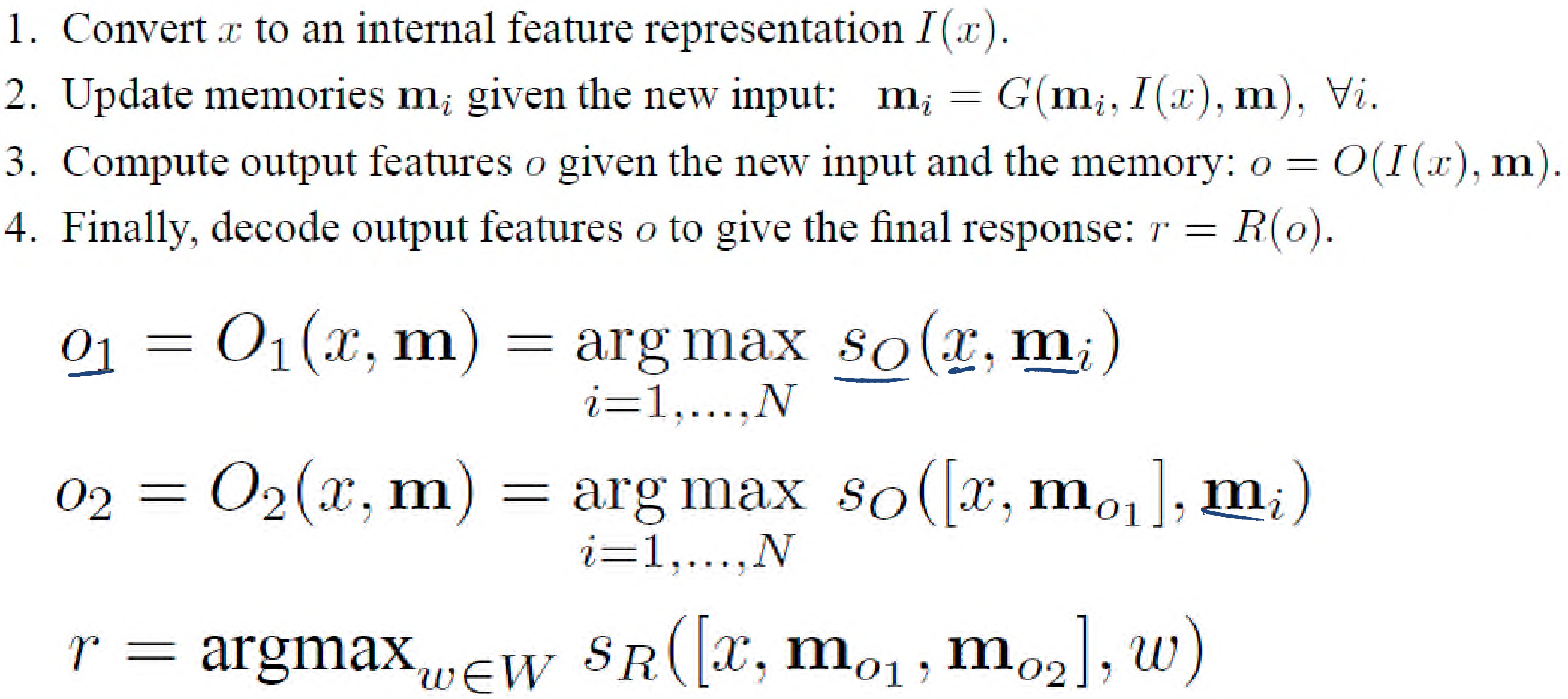
1. Convert x to an internal feature representation $I(x)$.
2. Update memories $\mathbf{m}_i$ given the new input: $\mathbf{m}_i = G(\mathbf{m}_i, I(x), \mathbf{m})$, $\forall i$.
3. Compute output features o given the new input and the memory: $o = O(I(x), \mathbf{m})$.
4. Finally, decode output features o to give the final response: $r = R(o)$.

“slot” in the memory: $\mathbf{m}_{S(x)} = I(x)$

next empty memory slot N: $\mathbf{m}_N = x$, $N = N + 1$

[Jason Weston, Sumit Chopra & Antoine Bordes, 2015]

Memory networks



1. Convert x to an internal feature representation $I(x)$.
2. Update memories $\mathbf{m}_i$ given the new input: $\mathbf{m}_i = G(\mathbf{m}_i, I(x), \mathbf{m}), \ \forall i$.
3. Compute output features o given the new input and the memory: $o = O(I(x), \mathbf{m})$.
4. Finally, decode output features o to give the final response: $r = R(o)$.

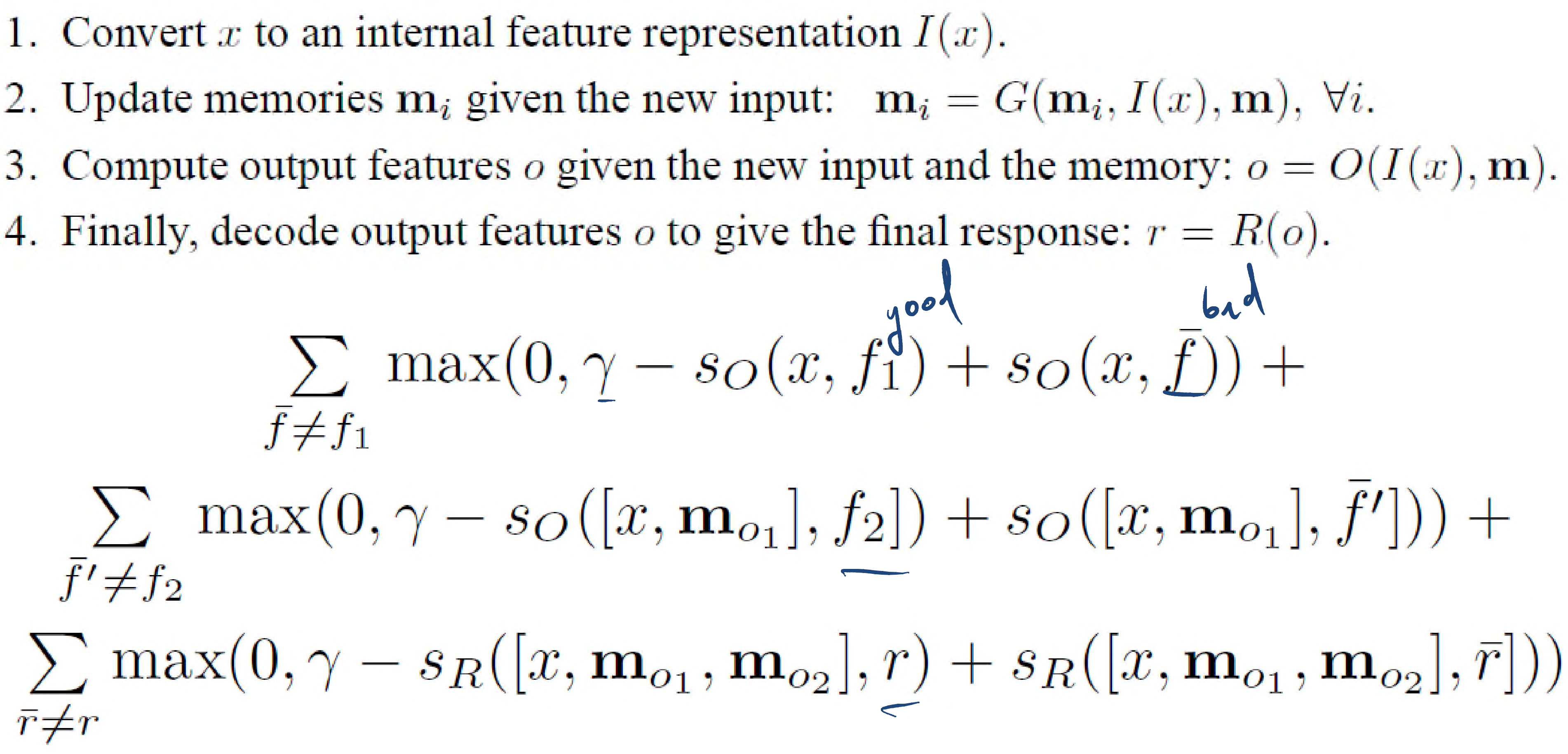
$$\underline{o_1} = O_1(x, \mathbf{m}) = \operatorname*{arg\,max}_{i=1,\dots,N} \ \underline{s_O}(\underline{x}, \underline{\mathbf{m}_i})$$

$$o_2 = O_2(x, \mathbf{m}) = \operatorname*{arg\,max}_{i=1,\dots,N} \ s_O([x, \mathbf{m}_{o_1}], \underline{\mathbf{m}_i})$$

$$r = \text{argmax}_{w \in W} \ s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], w)$$

[Jason Weston, Sumit Chopra & Antoine Bordes, 2015]

Memory networks



1. Convert x to an internal feature representation $I(x)$.
2. Update memories $\mathbf{m}_i$ given the new input: $\mathbf{m}_i = G(\mathbf{m}_i, I(x), \mathbf{m}),\ \forall i$.
3. Compute output features o given the new input and the memory: $o = O(I(x), \mathbf{m})$.
4. Finally, decode output features o to give the final response: $r = R(o)$.

$$\sum_{\bar{f} \neq f_1} \max(0, \underline{\gamma} - s_O(x, \overset{\text{good}}{f_1}) + s_O(x, \overset{\text{bad}}{\underline{\bar{f}}})) +$$

$$\sum_{\bar{f}' \neq f_2} \max(0, \gamma - s_O([x, \mathbf{m}_{o_1}], \underline{f_2}) + s_O([x, \mathbf{m}_{o_1}], \bar{f}')) +$$

$$\sum_{\bar{r} \neq r} \max(0, \gamma - s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], \underline{r}) + s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], \bar{r}]))$$

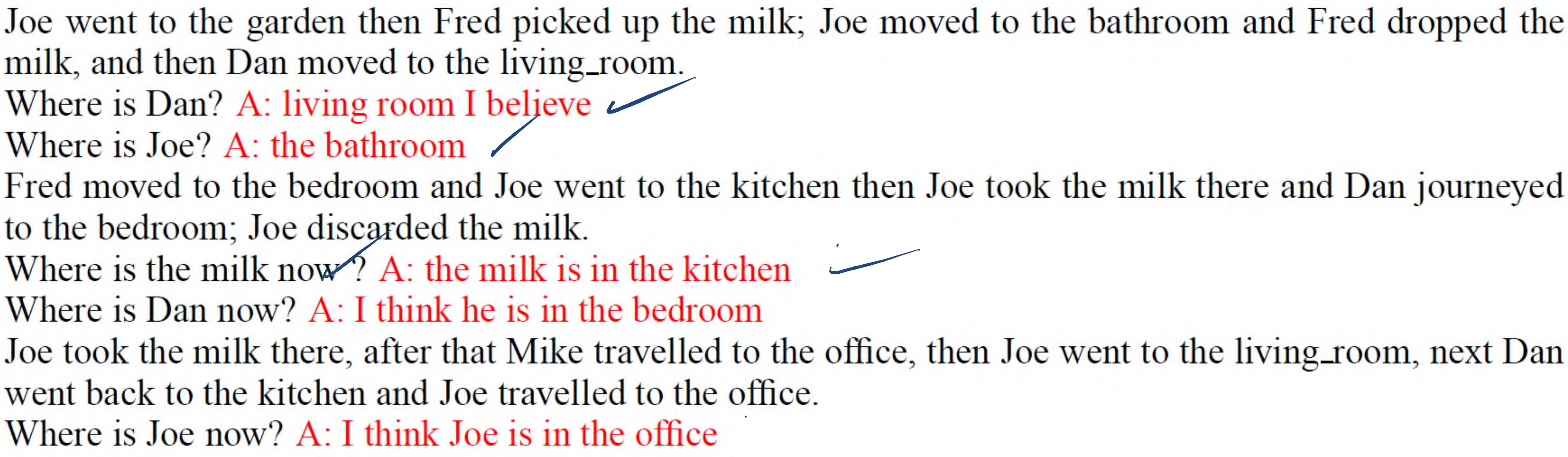
[Jason Weston, Sumit Chopra & Antoine Bordes, 2015]

Memory networks

$$
\begin{aligned}
&\sum_{\bar{f} \neq f_1} \max(0, \gamma - s_O(x, f_1) + s_O(x, \bar{f})) + \\
&\sum_{\bar{f}' \neq f_2} \max(0, \gamma - s_O([x, \mathbf{m}_{o_1}], f_2]) + s_O([x, \mathbf{m}_{o_1}], \bar{f}'])) + \\
&\sum_{\bar{r} \neq r} \max(0, \gamma - s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], r) + s_R([x, \mathbf{m}_{o_1}, \mathbf{m}_{o_2}], \bar{r}]))
\end{aligned}
$$

[Jason Weston, Sumit Chopra & Antoine Bordes, 2015]

Memory networks



Joe went to the garden then Fred picked up the milk; Joe moved to the bathroom and Fred dropped the milk, and then Dan moved to the living_room.
Where is Dan? A: living room I believe
Where is Joe? A: the bathroom
Fred moved to the bedroom and Joe went to the kitchen then Joe took the milk there and Dan journeyed to the bedroom; Joe discarded the milk.
Where is the milk now ? A: the milk is in the kitchen
Where is Dan now? A: I think he is in the bedroom
Joe took the milk there, after that Mike travelled to the office, then Joe went to the living_room, next Dan went back to the kitchen and Joe travelled to the office.
Where is Joe now? A: I think Joe is in the office

[Jason Weston, Sumit Chopra & Antoine Bordes, 2015]

Next lecture

In the next lecture, we will look models for sequences of data.