Sequence Labeling

Neural CRFs & Learning with Constraints

Mausam

Types of Prediction Tasks

- Two classes (**binary classification**)

  I hate this movie → positive / negative

  positive
  negative

- Multiple classes (**multi-class classification**)

  I hate this movie → very good / good / neutral / bad / very bad

  very good
  good
  neutral
  bad
  very bad

- Exponential/infinite labels (**structured prediction**)

  I hate this movie → PRP VBP DT NN

  I hate this movie → *kono eiga ga kirai*

Sequence problems

- Many problems in NLP have data which is a sequence of characters, words, phrases, lines, or sentences …
- We can think of our task as one of labeling each item

VBG	NN	IN	DT	NN	IN	NN
Chasing	opportunity	in	an	age	of	upheaval

POS tagging

B	B	I	I	B	I	B	I	B	B
而	相	对	于	这	些	品	牌	的	价

Word segmentation

PERS	O	O	O	ORG	ORG
Murdoch	discusses	future	of	News	Corp.

Named entity recognition

Q
A
Q
A
A
A
Q
A

Text segmentation

POS Tagging

DT NNP NN VBD VBN RP NN NNS
The Georgia branch had taken on loan commitments …

DT NN IN NN VBD NNS VBD
The average of interbank offered rates plummeted …

POS Tagging Ambiguity

- Words often have more than one POS: *back*
  - The *back* door = JJ
  - On my *back* = NN
  - Win the voters *back* = RB
  - Promised to *back* the bill = VB
- The POS tagging problem is to determine the POS tag for a particular instance of a word.

Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text, for example:
  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text, for example:
  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text, for example:
  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Person
Date
Location
Organi-
zation

The Named Entity Recognition Task

Task: Predict entities in a text

Foreign	ORG
Ministry	ORG
spokesman	O
Shen	PER
Guofang	PER
told	O
Reuters	ORG
:	O

Standard evaluation is per entity, *not* per token

Precision/Recall/F1 for IE/NER

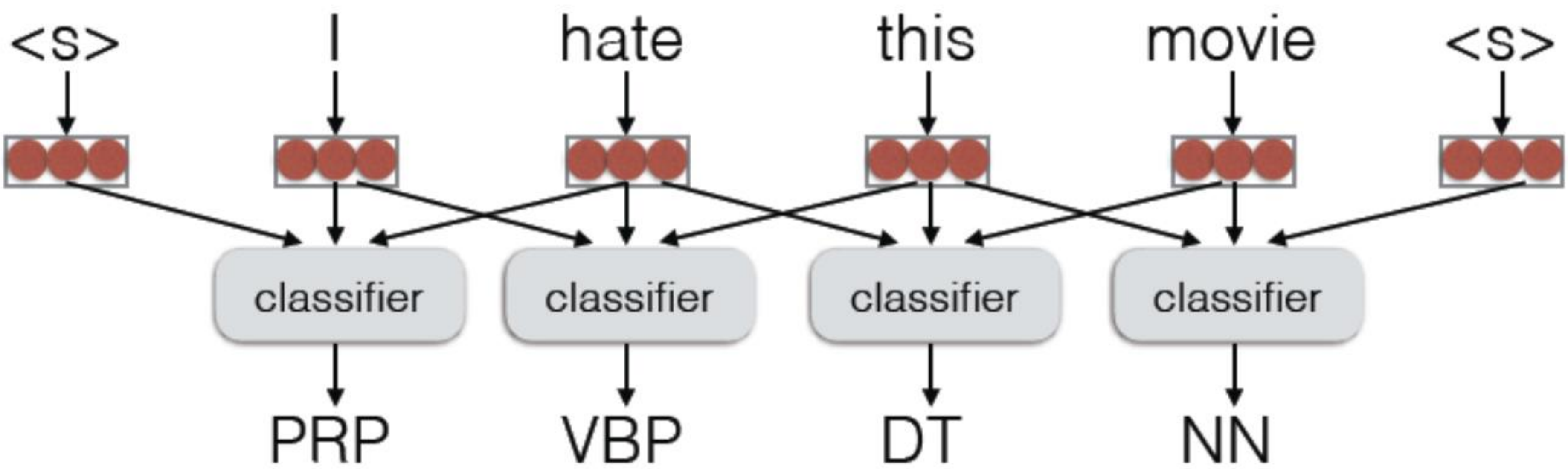
- Recall and precision are straightforward for tasks like IR and text categorization, where there is only one grain size (documents)
- The measure behaves a bit funnily for IE/NER when there are *boundary errors* (which are *common*):
  - First Bank of Chicago announced earnings …
- This counts as both a fp and a fn
- Selecting *nothing* would have been better
- Some other metrics (e.g., MUC scorer) give partial credit (according to complex rules)

Encoding classes for NER

	IO encoding	IOB encoding
Fred	PER	B-PER
showed	O	O
Sue	PER	B-PER
Mengqiu	PER	B-PER
Huang	PER	I-PER
‘s	O	O
new	O	O
painting	O	O

Practically negligible differences in performance. BIO is more standard..

Sequence Labeling as Independent Classification

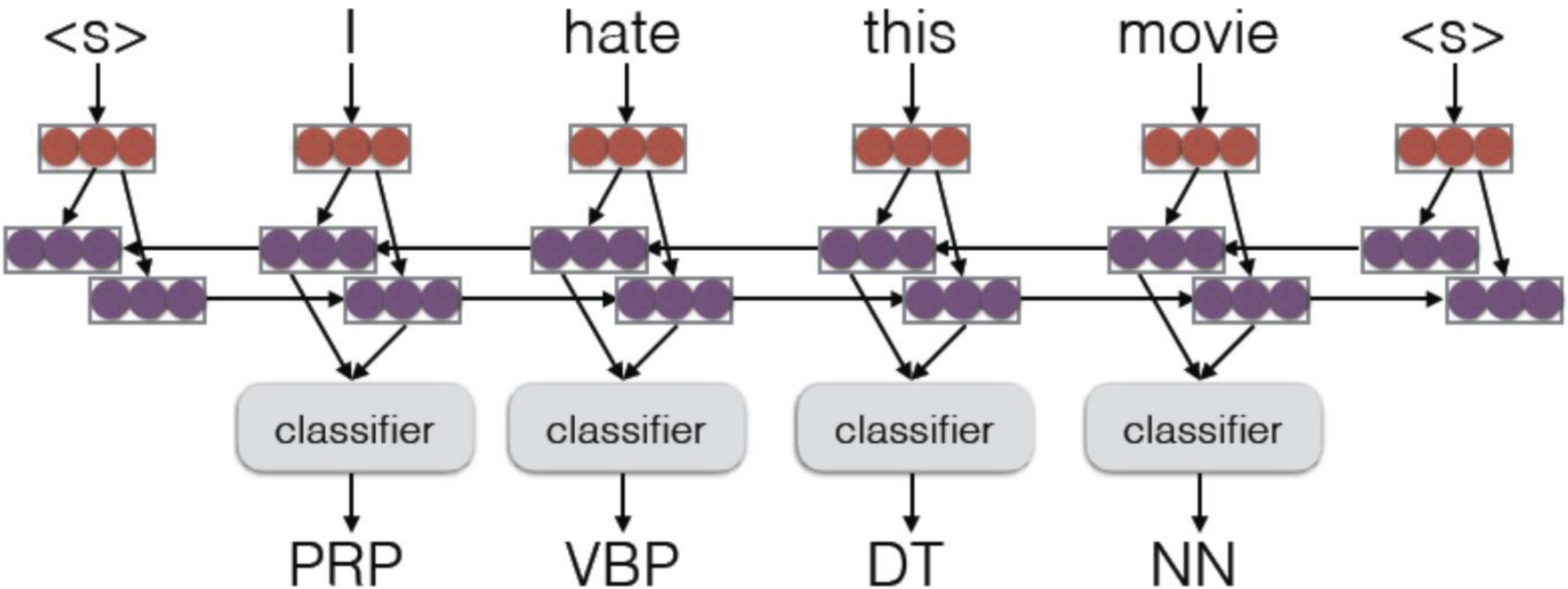


Structured Prediction task

But not a Structured Prediction Model

Instead: independent multi-class classification

Sequence Labeling with BiLSTM / Transformer



What is missing?

Still not modeling output structure!

Outputs are independent (of each other)

Why Model Interactions in Output?

- Consistency is important!

time	flies	like	an	arrow	
NN	VBZ	IN	DT	NN	(time moves similarly to an arrow)
NN	NNS	VB	DT	NN	("time flies" are fond of arrows)
VB	NNS	IN	DT	NN	(please measure the time of flies similarly to how an arrow would)
NN	NNS	IN	DT	NN	("time flies" that are similar to an arrow)

- Example 2: Paris Hilton

Conditional Random Fields

- Models w/ Local Dependencies

- Some independence assumptions on the output space, but not entirely independent (local dependencies)

- Exact and optimal decoding/training via dynamic programs

Local vs Global Normalization

- **Locally normalized models:** each decision made by the model has a probability that adds to one

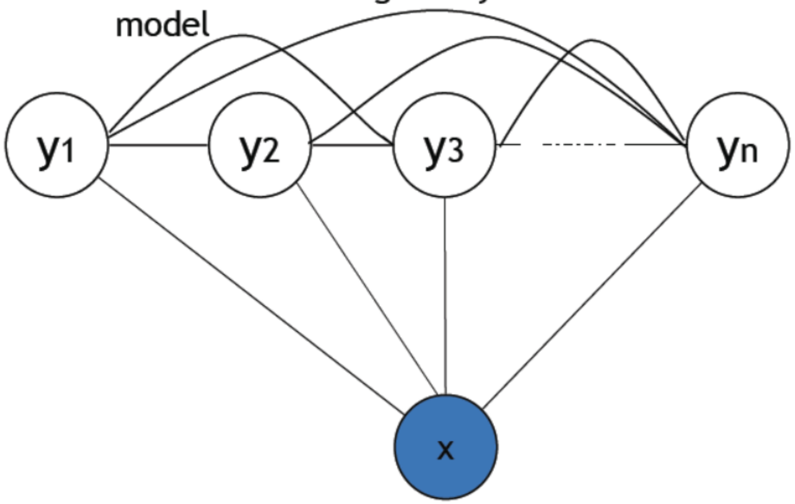
$$P(Y \mid X) = \prod_{j=1}^{|Y|} \frac{e^{S(y_j|X,y_1,\ldots,y_{j-1})}}{\sum_{\tilde{y}_j \in V} e^{S(\tilde{y}_j|X,y_1,\ldots,y_{j-1})}}$$

- **Globally normalized models (a.k.a. energy-based models):** each sequence has a score, which is not normalized over a particular decision

$$P(Y \mid X) = \frac{e^{\sum_{j=1}^{|Y|} S(y_j|X,y_1,\ldots,y_{j-1})}}{\sum_{\tilde{Y} \in V*} e^{\sum_{j=1}^{|\tilde{Y}|} S(\tilde{y}_j|X,\tilde{y}_1,\ldots,\tilde{y}_{j-1})}}$$

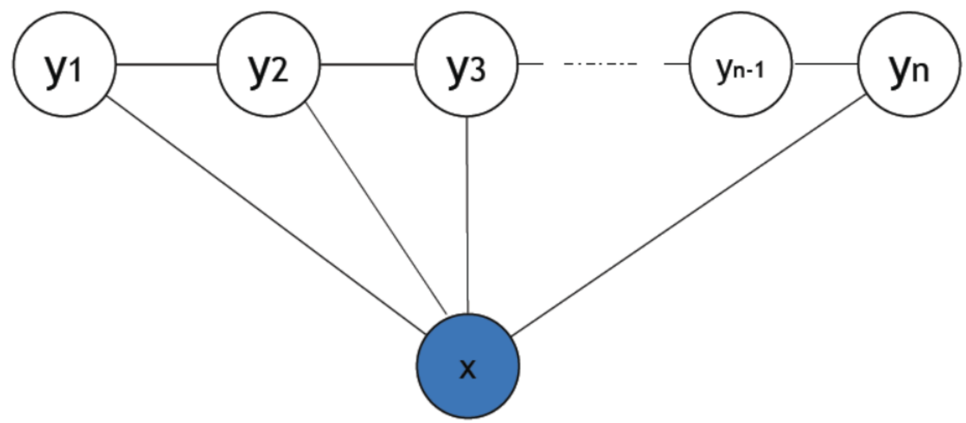
CRFs

General form of globally normalized model



$$P(Y|X) = \frac{\psi(Y,X)}{\sum_{Y'} \psi(Y',X)}$$

First-order linear CRF



$$P(Y|X) = \frac{\prod_{i=1}^{L} \psi_i(y_{i-1}, y_i, X)}{\sum_{Y'} \prod_{i=1}^{L} \psi_i(y'_{i-1}, y'_i, X)}$$

Potential Functions

- $\psi_i(y_{i-1}, y_i, X) = \exp\Big(\underbrace{\boxed{W^T T(y_{i-1}, y_i, X, i)}}_{\text{"Transition"}} + \underbrace{\boxed{U^T\ S(y_i, X, i)}}_{\text{"Emission"}} + b_{y_{i-1}, y_i}\Big)$
  - Using neural features in DNN:
    $$\psi_i(y_{i-1}, y_i, X) = \exp\left(W_{y_{i-1},y_i}^T F(X, i) + U_{y_i}^T\ F(X, i) + b_{y_{i-1},y_i}\right)$$
    - Number of parameters: $O(|Y|^2 d_F)$
  - Simpler version:
    $$\psi_i(y_{i-1}, y_i, X) = \exp\left(W_{y_{i-1},y_i} + U_{y_i}^T F(X, i) + b\right)$$
    - Number of parameters: $O(|Y|^2 + |Y| d_F)$

Linear Chain CRF (in practice)

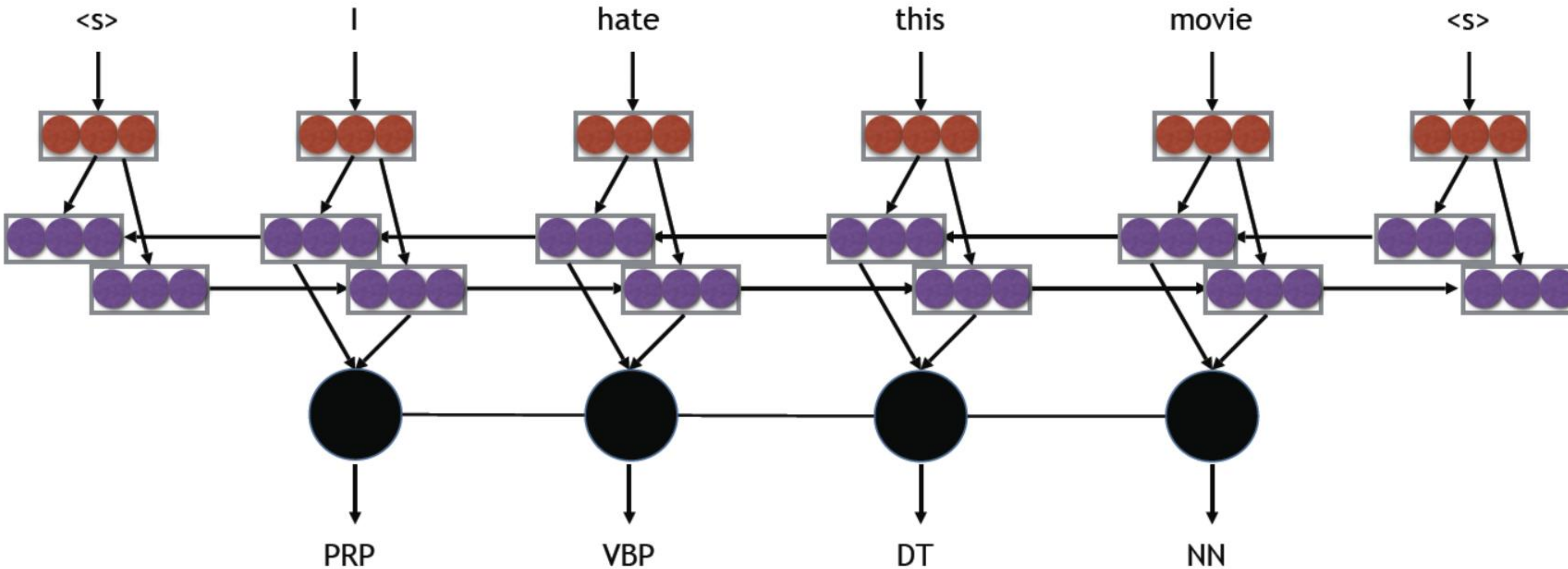
$$\psi_i(y_{i-1}, y_i, X) = \exp\left(W_{y_{i-1}, y_i} + U_{y_i}^T F(X, i) + b\right)$$

- Score(X,Y) = $\sum_{i=1}^{T+1} W_{[y_{i-1}, y_i]} + \sum_{i=1}^{T} e(x_i, y_i)$

- For a tagset of K possible tags,
  - introduce a scoring matrix W ϵ R^{KxK} in which
  - W[g,h]= compatibility score of the tag sequence g h.

- Global inference

BiLSTM-CRF



Properties

$$Z(X) = \sum_{Y} \prod_{i=1}^{L} \psi_i(y_{i-1}, y_i, X)$$

- Each label depends on the input, and the nearby labels
- But given *adjacent* labels, others do not matter
- If we knew the score of every sequence $y_1, \ldots, y_{n-1}$, we could compute easily the score of sequence $y_1, \ldots, y_{n-1}, y_n$
- So we really only need to know the score of all the sequences ending in each y_{n-1}
- Think of that as some "precalculation" that happens before we think about y_n

Decoding Problem

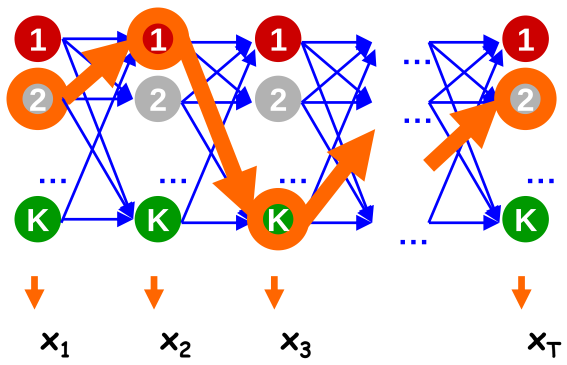
Given $X=x_1 \ldots x_T$, what is "best" tagging $y_1 \ldots y_T$?

Several possible meanings of 'solution'

1. States which are individually most likely
2. Single best state sequence

We want ***sequence*** $y_1 \ldots y_T$,
such that $P(Y|X)$ is maximized

$$Y^* = \text{argmax}_Y P(\, Y|X\,)$$



Most Likely Sequence

- Problem: find the most likely (Viterbi) sequence under the model
  - Given model parameters, we can score any sequence pair

NNP	VBZ	NN	NNS	CD	NN	.
Fed	raises	interest	rates	0.5	percent	.

  - In principle, we're done – list all possible tag sequences, score each one, pick the best one (the Viterbi state sequence)

2T+1 operations per sequence

NNP VBZ NN NNS CD NN ⇨ logP = -23

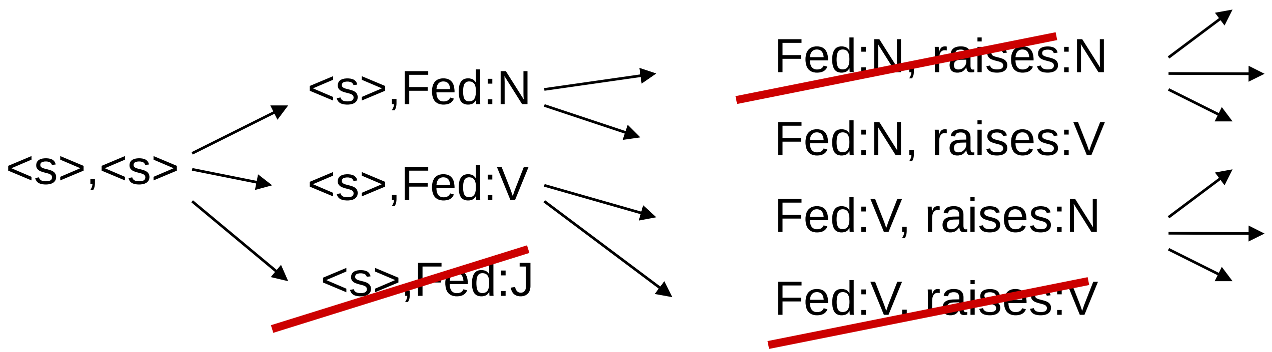
NNP NNS NN NNS CD NN ⇨ logP = -29

NNP VBZ VB NNS CD NN ⇨ logP = -27

$|Y|^T$ tag sequences!

Finding the Best Trajectory

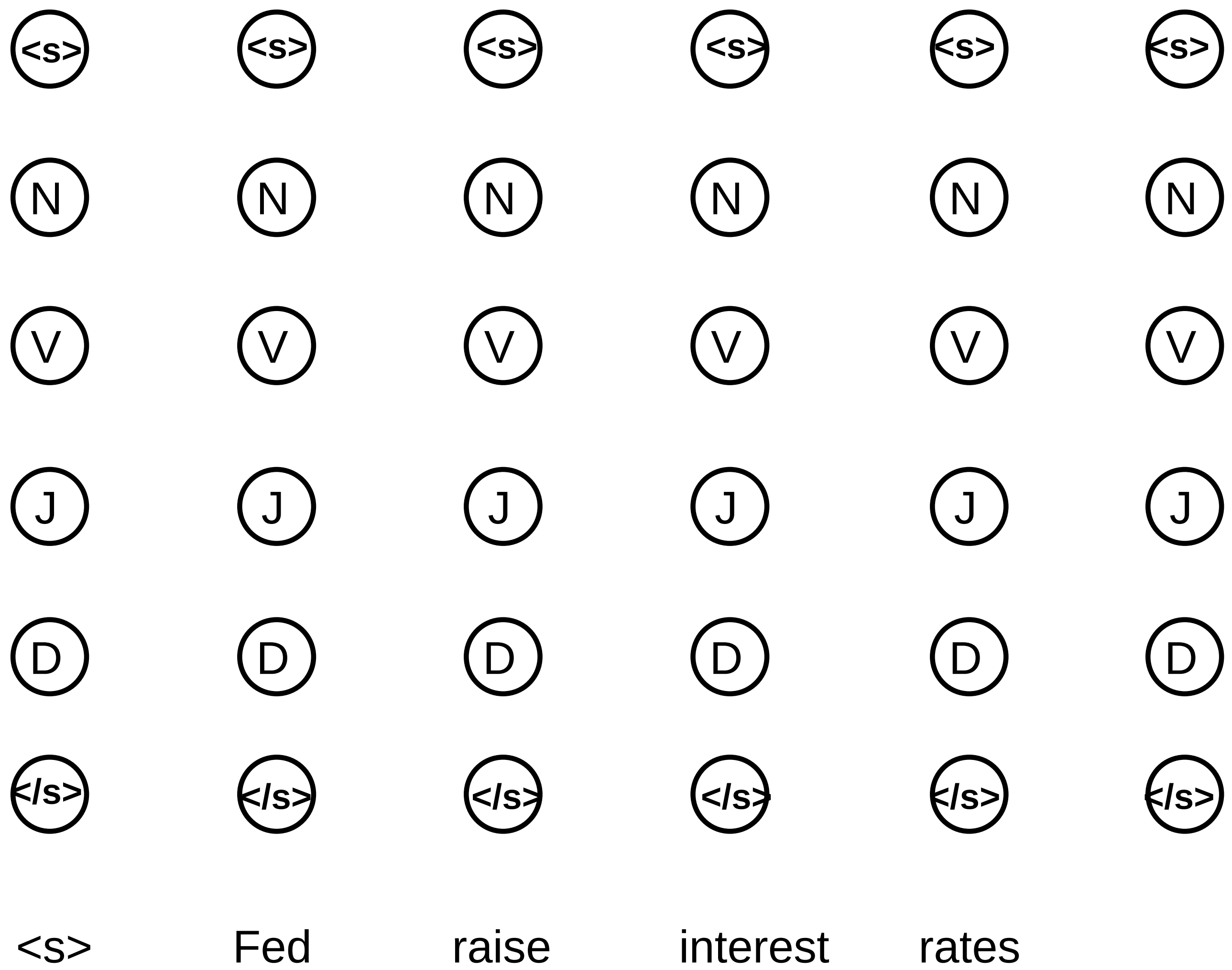
- Brute Force: Too many trajectories (state sequences) to list
- Option 1: Beam Search



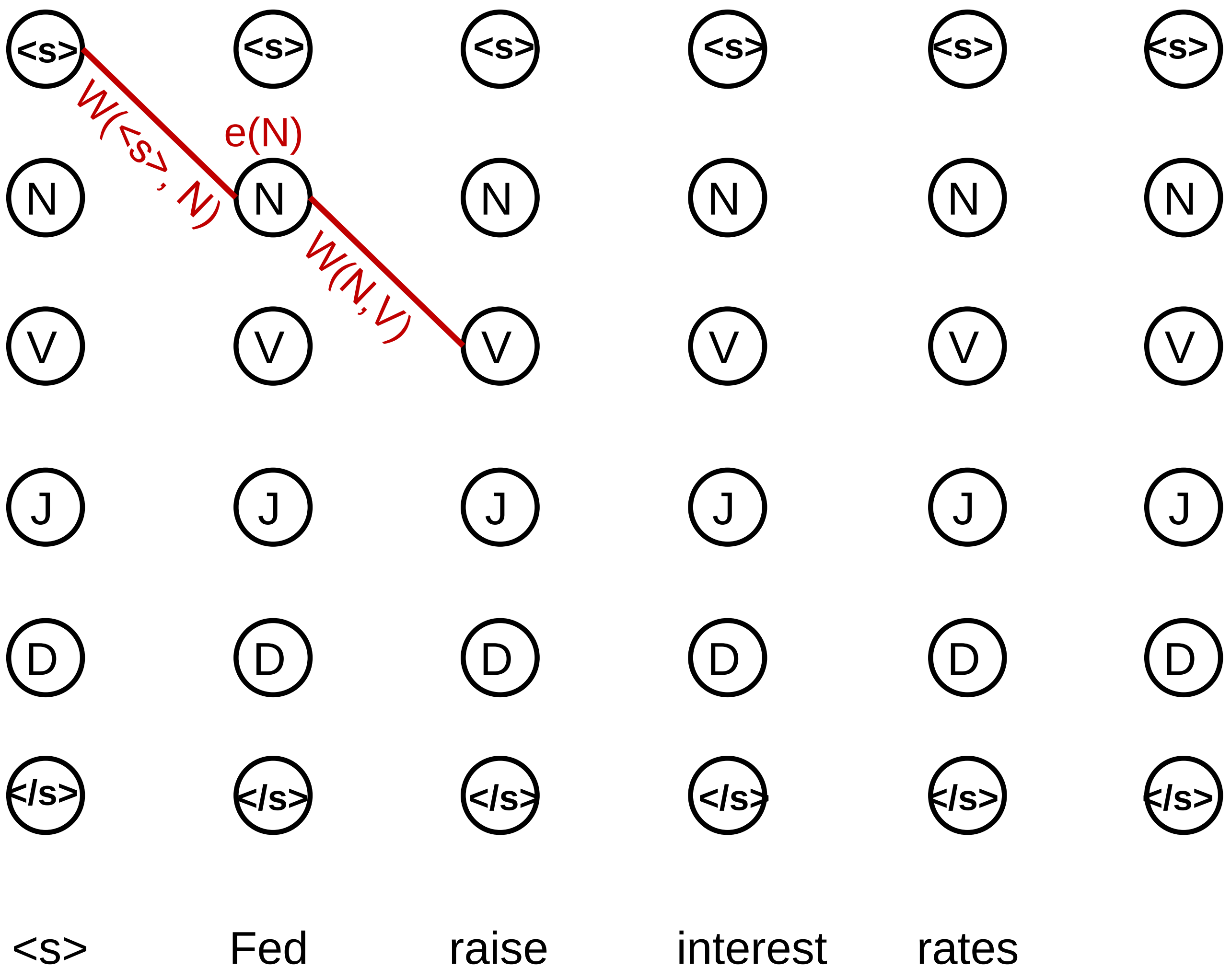
  - A beam is a set of partial hypotheses
  - Start with just the single empty trajectory
  - At each derivation step:
    - Consider all continuations of previous hypotheses
    - Discard most, keep top k

- Beam search works ok in practice
  - … but sometimes you want the optimal answer
  - … and there’s often a better option than naïve beams

State Lattice / Trellis



State Lattice / Trellis



Dynamic Programming

- Decoding: $$Y^* = \arg\max_Y P(Y \mid X) = \arg\max_Y score(X, Y)$$ $$= \arg\max_Y \sum_{t=1}^{T+1} W(y_{t-1}, y_t) + \sum_{t=1}^{T} e(X, y_t)$$
- First consider how to compute max
- Define $\delta_i(y_i) = \max_{y[1:i-1]} score(X, y_{[1..i]})$
  - score of ***most likely*** label sequence ending with tag y_i at position i, given words $x_1, \ldots, x_T$

$$\delta_i(y_i) = \max_{y[1:i-1]} e(X, y_i) + W(y_{i-1}, y_i) + score(X, y_{[1..i-1]})$$

$$= e(X, y_i) + \max_{y_{i-1}} W(y_{i-1}, y_i) + \max_{y[1:i-2]} score(X, y_{[1..i-1]})$$

$$= e(X, y_i) + \max_{y_{i-1}} W(y_{i-1}, y_i) + \delta_{i-1}(y_{i-1})$$

Viterbi Algorithm

- Input: $x_1,\ldots,x_T$, W() and e()
- Initialize: $\delta_0(\langle s \rangle) = 0$, and –infinity for other labels
- For i=1 to T do
  - For (y') in all possible tagset

$$\delta_i(y') = e(X, y') + \max_y W(y, y') + \delta_{i-1}(y)$$

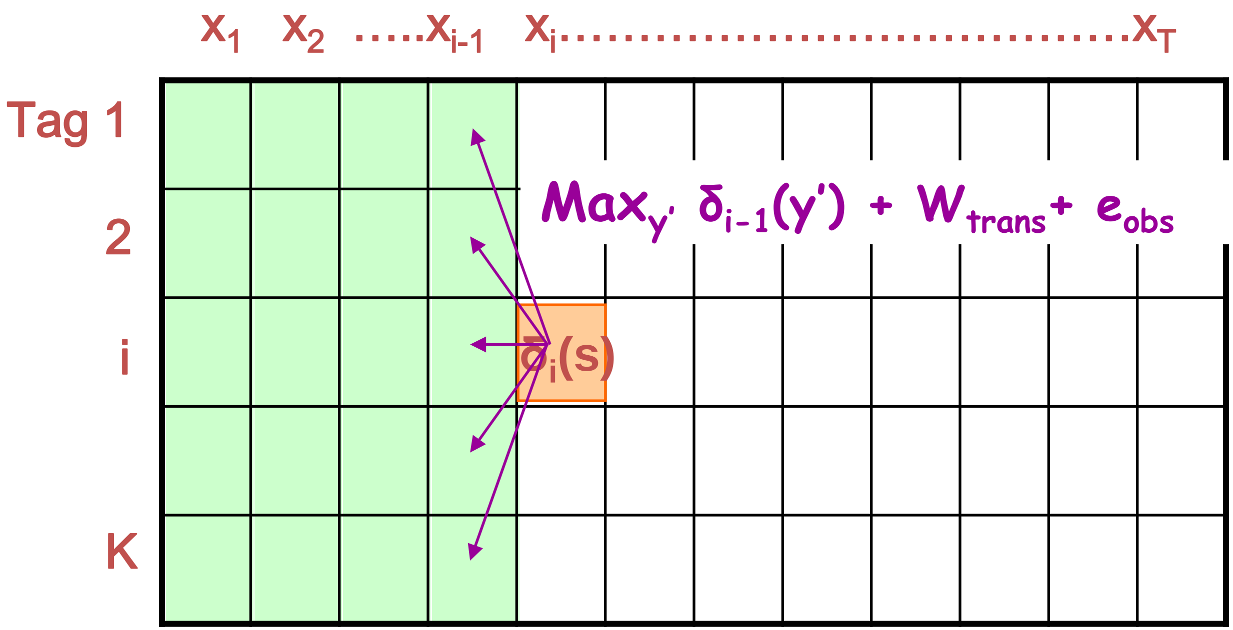
- Return

$$\max_{y'} W(y', </s>) + \delta_T(y')$$

returns only the optimal value

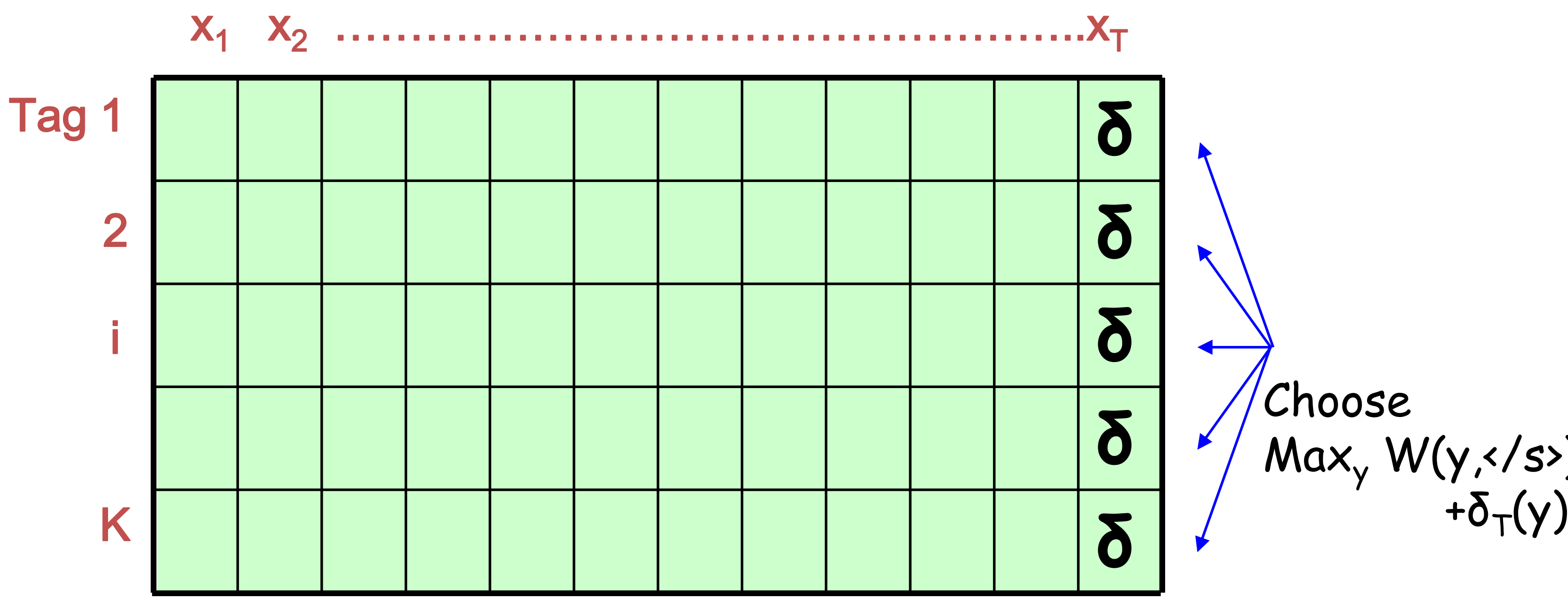
keep backpointers

Viterbi Algorithm

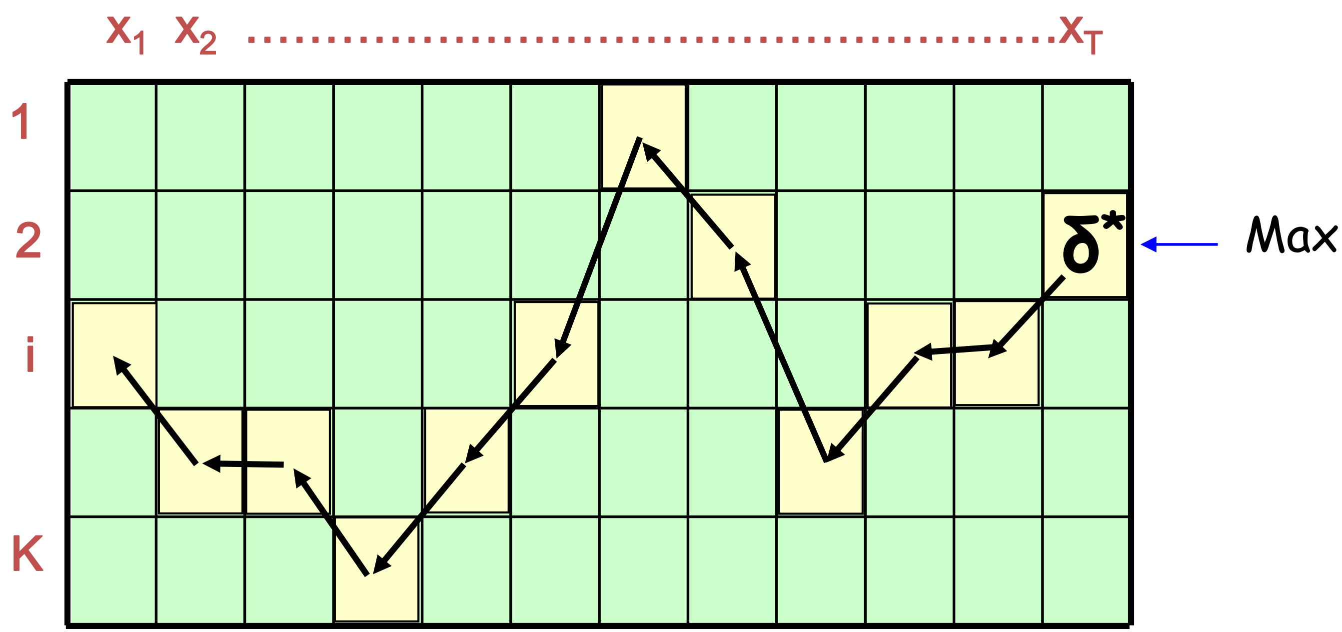


Remember: $\delta_i(y)$ = score of most likely tag seq ending with y at time i

Terminating Viterbi



Terminating Viterbi



How did we compute δ*?

$\text{Max}_{s'}\ \delta_{T-1}(y') + P_{trans} + P_{obs}$

Now Backchain to Find Final Sequence

Time: $O(|Y|^2T)$

Space: $O(|Y|T)$

Linear in length of sequence

Training

- Find weights such that

$$Loss(\theta) = -\log P_{CRF}(Y \mid X;\theta)$$

is maximized

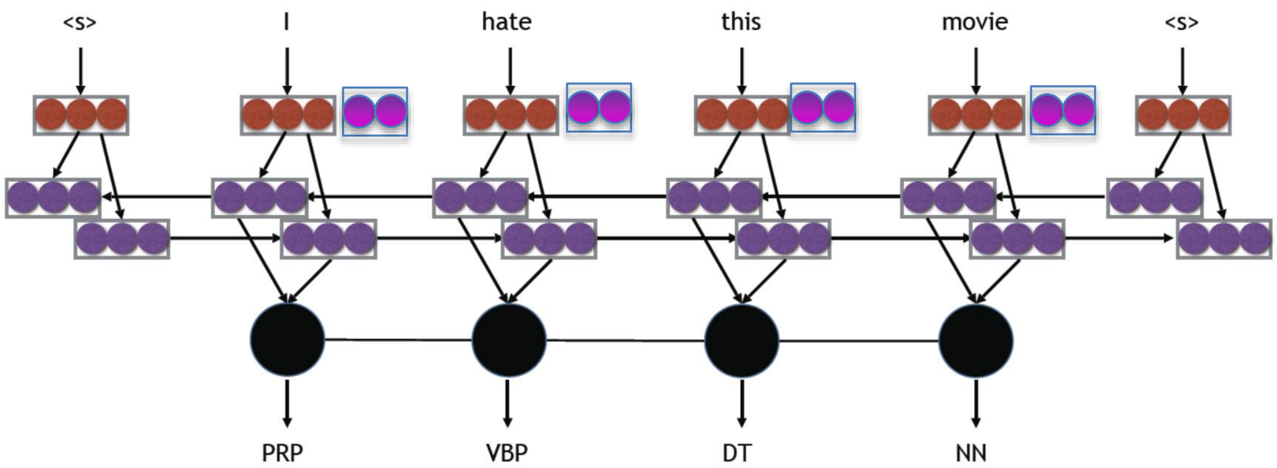
$$P(Y|X) = \frac{\prod_{i=1}^{L} \psi_i(y_{i-1}, y_i, X)}{\sum_{Y'} \prod_{i=1}^{L} \psi_i(y'_{i-1}, y'_i, X)}$$

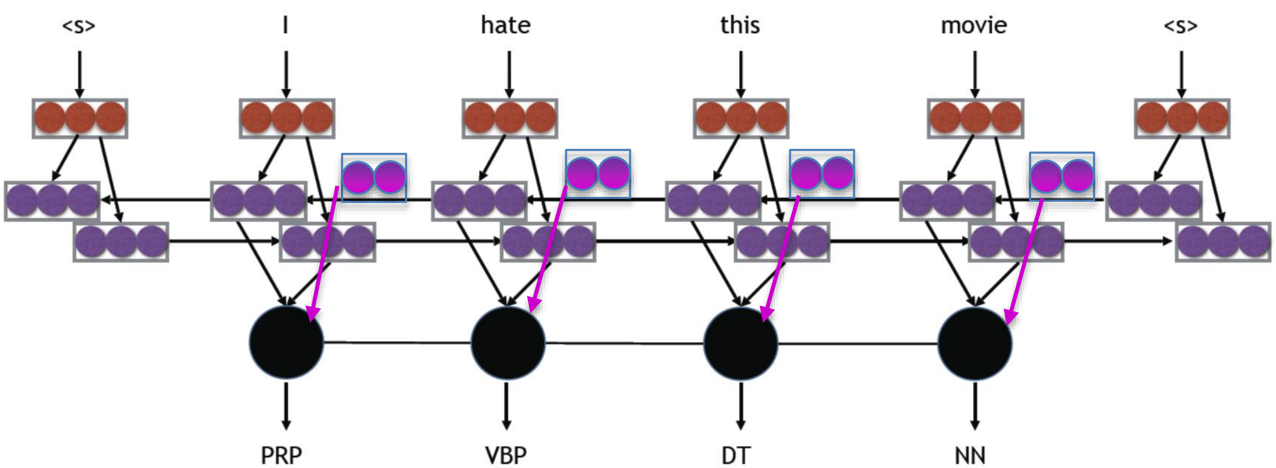
Log_sum_exp
(additive terms)

How to compute partition function?

(backward step handled by autograd)

BiLSTM-CRF w/ Features





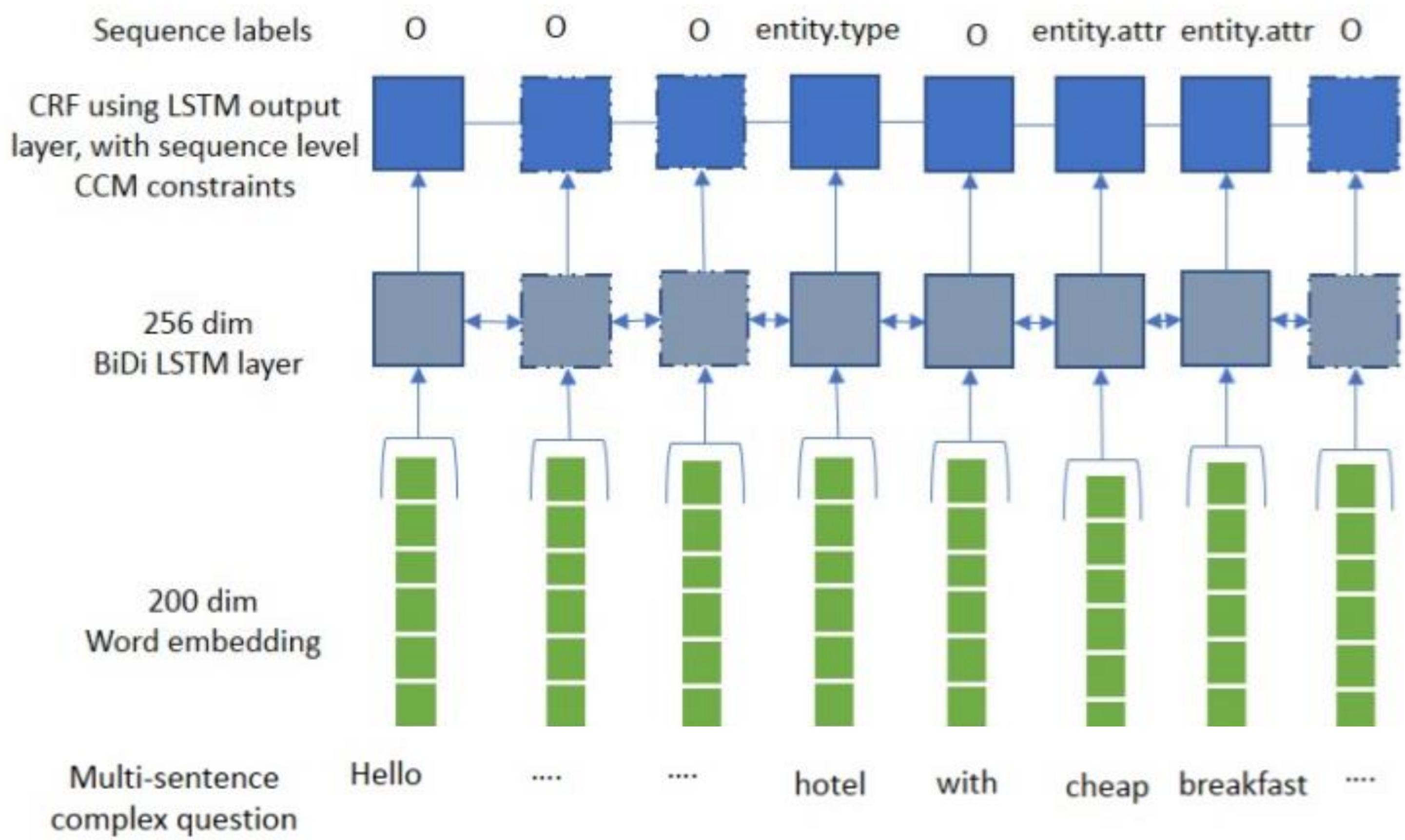
MSQU: Multi-Sentence Qn Understanding

- *“I am taking 15 Scouts to New Zealand over Christmas and New Year. We are spending NYE in Auckland and are looking for suggestions of restaurants (maybe buffet style) which will be suitable for a large group? Ideally close to somewhere where we can watch the fireworks from. Any ideas would be welcome”*

~Open Question Understanding

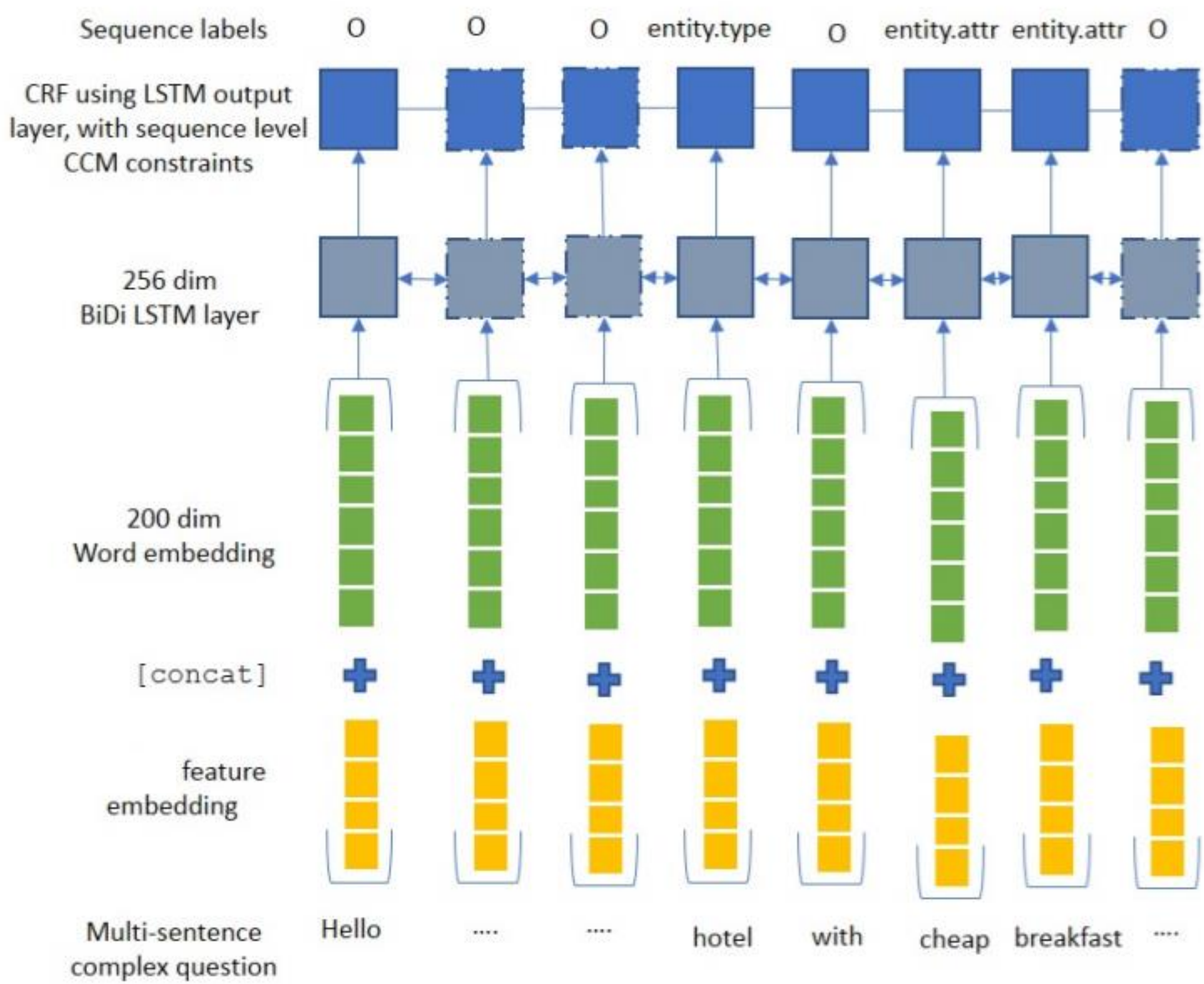
select x where x.type = “restaurant” and
x.location IN “Auckland” and x.attribute = “buffet style” and
x.attribute = “suitable for large group” and
x.attribute PREF “somewhere we can watch fireworks from”

Key Issue: Only 150 labeled questions!


Sequence labels
O
O
O
entity.type
O
entity.attr
entity.attr
O
CRF using LSTM output layer, with sequence level CCM constraints
256 dim
BiDi LSTM layer
200 dim
Word embedding
Multi-sentence complex question
Hello
....
....
hotel
with
cheap
breakfast
....

Human Insight: Features!

- Token level features
  - Raw token, lexicalized features, POS Tag, NER Tags
- Hand designed features
  - Indicator features for candidates that are likely to be types based on targets of WH- POS words such as Which, Where etc
  - Indicator features for candidates that are likely to be attributes by checking if there is an edge in the dependency graph leading up to a candidate type.
  - Indicator features for adj-noun phrases
- Cluster ids of word2vec clustered words
- Global word counts in post


Sequence labels
O
O
O
entity.type
O
entity.attr
entity.attr
O
CRF using LSTM output layer, with sequence level CCM constraints
256 dim BiDi LSTM layer
200 dim Word embedding
[concat]
feature embedding
Multi-sentence complex question
Hello
....
....
hotel
with
cheap
breakfast
....

Question Parsing Accuracy

[Contractor, Patra, Mausam, Singla JNLE'21]

Model	F1 (type)	F1 (attribute)	F1 (location)	F1 (macro-avg)
CRF (with Features)	51.4	45.3	55.7	50.8
BiLSTM CRF	53.3	47.6	52.1	51.0
BiLSTM CRF + Features	**58.4**	**48.1**	**62.0**	**56.2**

Neural + Features > Neural > Symbolic + Features

Question Parsing Accuracy

[Contractor, Patra, Mausam, Singla JNLE'21]

Model	F1 (type)	F1 (attribute)	F1 (location)	F1 (macro-avg)
CRF	51.4	45.3	55.7	50.8
BiLSTM CRF	53.3	47.6	52.1	51.0
BERT	59.6	50.6	59.5	56.6
BERT + BiLSTM + CRF	**63.4**	**56.5**	**72.4**	**64.4**

BERT + CRF > BERT

Summary

- BiLSTM+CRF
  - combines feature engineering of LSTMs
  - global reasoning of CRFs

- When are CRFs helpful?
  - Joint inference
  - Low data setting

Deep Learning With Constraints

Yatin Nandwani

Work done in collaboration with

Abhishek Pathak

Under the guidance of

Prof. Mausam and *Prof.* Parag Singla

Learning with Constraints: *Motivation*

➔ Modern day AI == Deep Learning (DL) [Learn from Data]

Learning with Constraints: *Motivation*

- ➔ Modern day AI == Deep Learning (DL) [Learn from Data]
- ➔ Can we inject symbolic knowledge in Deep Learning? E.g. Person => Noun [Learn from ~~Data~~ Knowledge]*(credit: Vivek S Kumar)*

Learning with Constraints: *Motivation*

→ Modern day AI == Deep Learning (DL) [Learn from Data]

→ Can we inject symbolic knowledge in Deep Learning? E.g. Person => Noun [Learn from ~~Data~~ Knowledge]

→ **Constraints:** One of the ways of representing symbolic knowledge.

$$\mathbb{1}\{y_{PER.} = 1\} \implies \mathbb{1}\{y_{Noun.} = 1\}$$

Learning with Constraints: *Motivation*

- **Task:** Fine Grained Entity Typing

Learning with Constraints: *Motivation*

Input: Bag of Mentions

Sample Mention: "*Barack Obama is the President of the United States*"

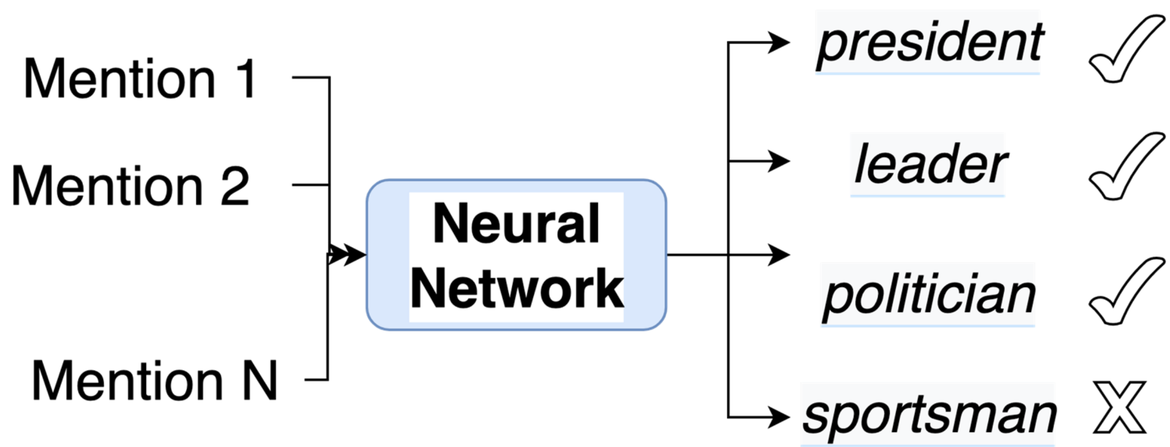
Output: *president, leader, politician...*

Learning with Constraints: *Motivation*

Input: Bag of Mentions

Sample Mention: "*Barack Obama is the President of the United States*"

Output: *president, leader, politician...*

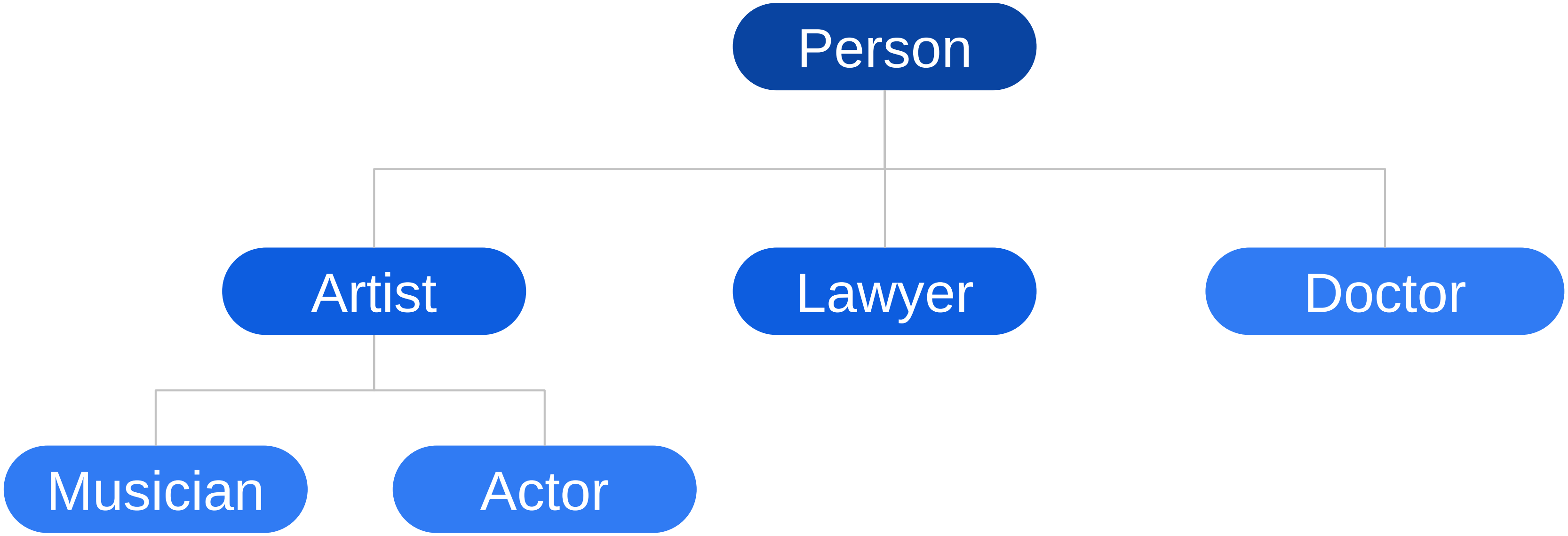


Learning with Constraints: *Motivation*

- **Constraints:** Hierarchy on Output label space

Learning with Constraints: *Motivation*

- **Constraints:** Hierarchy on Output label space



Learning with Constraints: *Motivation*

→ **Using Soft Logic**

$$\mathbb{1}\left\{y_{ARTIST} = 1\right\} \implies \mathbb{1}\left\{y_{PERSON} = 1\right\}$$

Learning with Constraints: *Motivation*

→ Using Soft Logic

$$\mathbb{1}\left\{y_{ARTIST}=1\right\} \implies \mathbb{1}\left\{y_{PERSON}=1\right\}$$

$$\left(\neg\mathbb{1}\left\{y_{ARTIST}=1\right\}\right) \vee \left(\mathbb{1}\left\{y_{PERSON}=1\right\}\right)$$

Learning with Constraints: *Motivation*

→ **Using Soft Logic**

$$\mathbb{1}\{y_{ARTIST} = 1\} \implies \mathbb{1}\{y_{PERSON} = 1\}$$

$$(\neg \mathbb{1}\{y_{ARTIST} = 1\}) \vee (\mathbb{1}\{y_{PERSON} = 1\})$$

$$(1 - p(y_{ARTIST})) + p(y_{PERSON})$$

Learning with Constraints:

Re

→ l

Boolean Expression	T-norm: Choice 1	T-norm: Choice 2
v	$p(v=1)$	
$\neg v$	$1-p(v=1)$	
$v_1 \vee v_2$	$\min(p(v_1=1)+p(v_2=1),1)$	$\max(p(v_1=1),p(v_2=1))$
$v_1 \wedge v_2$	$\max(p(v_1=1)+p(v_2=1)-1,0)$	$\min(p(v_1=1),p(v_2=1))$

$$\mathbb{1}\{y_{ARTIST}=1\} \implies \mathbb{1}\{y_{PERSON}=1\}$$

$$(\neg\mathbb{1}\{y_{ARTIST}=1\}) \vee (\mathbb{1}\{y_{PERSON}=1\})$$

$$(1-p(y_{ARTIST}))+p(y_{PERSON})$$

Learning with Constraints: *Representation of Constraints*

$$1 - p(y_{ARTIST}) + p(y_{PERSON}) = 1$$

Learning with Constraints: *Representation of Constraints*

$$1 - p(y_{ARTIST}) + p(y_{PERSON}) = 1$$

$$1 - p(y_{ARTIST}) + p(y_{PERSON}) \geq 1$$

Learning with Constraints: *Representation of Constraints*

$$1 - p(y_{ARTIST}) + p(y_{PERSON}) = 1$$

$$1 - p(y_{ARTIST}) + p(y_{PERSON}) \geq 1$$

Equivalently:

$$p(y_{ARTIST}) - p(y_{PERSON}) \leq 0$$

Learning with Constraints: *Representation of Constraints*

Define:

$$f_k^i = p(y_{ARTIST}) - p(y_{PERSON})$$

kth Constraint

ith Data point

Inequality Constraint:

$$f_k^i \leq 0$$

Learning with Constraints: *Formulation*

Unconstrained Problem

$$\min_w L(w)$$

$L(w)$: Any standard loss function, say Cross Entropy

Learning with Constraints: *Formulation*

Unconstrained Problem

$$\min_{w} L(w)$$

$L(w)$: Any standard loss function, say Cross Entropy

Constrained Problem

$$\min_{w} L(w) \quad \text{subject to} \quad f_k^i(w) \leq 0; \ \ \forall 1 \leq i \leq m; \ \ \forall 1 \leq k \leq K$$

Learning with Constraints: *Formulation*

Constrained Problem

$$\min_{w} L(w) \quad \text{subject to} \quad f_k^i(w) \leq 0; \;\; \forall 1 \leq i \leq m; \;\; \forall 1 \leq k \leq K$$

Where:

m: Size of training data

K: Number of Constraints

Learning with Constraints: *Formulation*

Constrained Problem

$$\min_{w} L(w) \quad \text{subject to} \quad f_k^i(w) \le 0; \;\; \forall 1 \le i \le m; \;\; \forall 1 \le k \le K$$

Lagrangian

$$\mathcal{L}(w, \Lambda) = L(w) + \sum_{i=1}^{m} \sum_{k=1}^{K} \lambda_k^i f_k^i(w)$$

Learning with Constraints: *Formulation*

Constrained Problem

$$\min_{w} L(w) \quad \text{subject to} \quad f_k^i(w) \leq 0; \;\; \forall 1 \leq i \leq m; \;\; \forall 1 \leq k \leq K$$

Where:

m: Size of training data

K: Number of Constraints

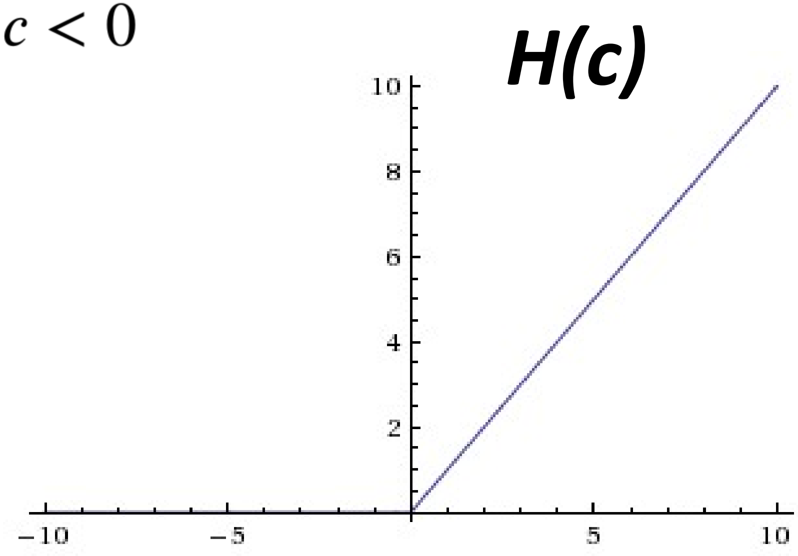
Issue:

O(mK) #constraints

i.e. ***mK*** Lagrange Multipliers!

Learning with Constraints: *Reduce # Constraints*

$$H(c) = c \quad for \quad c \geq 0, \quad \text{and} \quad 0 \quad \text{for} \quad c < 0$$

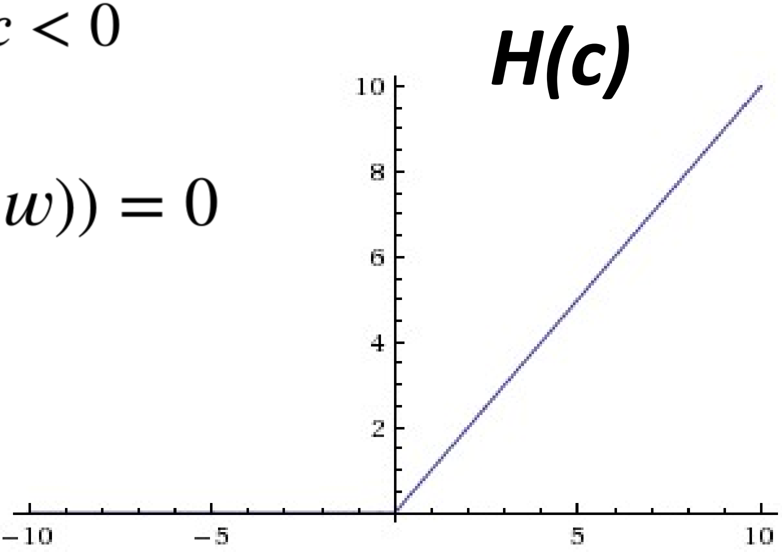


Learning with Constraints: *Reduce # Constraints*

$$H(c) = c \quad for \quad c \geq 0, \quad \text{and} \quad 0 \quad \text{for} \quad c < 0$$

$$f_k^i(w) \leq 0 \qquad \equiv \qquad H(f_k^i(w)) = 0$$

Equivalent



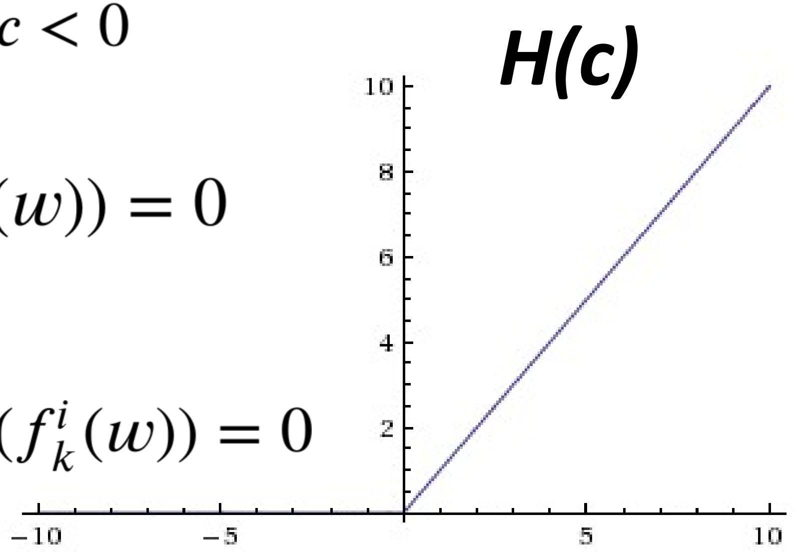
Learning with Constraints: *Reduce # Constraints*

$$H(c) = c \;\; for \;\; c \geq 0, \quad \text{and} \quad 0 \quad \text{for} \quad c < 0$$

$$f_k^i(w) \leq 0 \qquad \equiv \qquad H(f_k^i(w)) = 0$$

Equivalent

$$\forall i : H(f_k^i(w)) = 0 \qquad \equiv \qquad \sum_i H(f_k^i(w)) = 0$$



Learning with Constraints: *Reduce # Constraints*

Originally:

$$\min_{w} L(w) \quad \text{subject to} \quad f_k^i(w) \leq 0; \;\; \forall 1 \leq i \leq m; \;\; \forall 1 \leq k \leq K$$

Learning with Constraints: *Reduce # Constraints*

Originally:

$$\min_w L(w) \quad \text{subject to} \quad f_k^i(w) \le 0; \;\; \forall 1 \le i \le m; \;\; \forall 1 \le k \le K$$

Now:

Define: $h_k(w) = \sum_i H(f_k^i(w))$

$$\min_w L(w) \quad \text{subject to} \quad h_k(w) = 0; \;\; \forall 1 \le k \le K$$

Learning with Constraints: *Reduce # Constraints*

Originally:

$$\min_w L(w) \quad \text{subject to} \quad f_k^i(w) \leq 0; \;\; \forall 1 \leq i \leq m; \;\; \forall 1 \leq k \leq K$$

Now:

Define: $h_k(w) = \sum_i H(f_k^i(w))$

O(K) #constraints

$$\min_w L(w) \quad \text{subject to} \quad h_k(w) = 0; \;\; \forall 1 \leq k \leq K$$

Learning with Constraints: *Reduce # Constraints*

$$\min_{w} L(w) \quad \text{subject to} \quad h_k(w) = 0; \;\; \forall 1 \leq k \leq K$$

Lagrangian

$$\mathcal{L}(w; \Lambda) = L(w) + \sum_{k=1}^{K} \lambda_k h_k(w)$$

Learning with Constraints: *Experiments:* Typenet

	MAP Scores			Constraint Violations		
Scenario	**5% Data**	**10% Data**	**100% Data**	**5% Data**	**10% Data**	**100% Data**
B	68.6			22,715		
B+H	68.71			22,928		
B+C						
B+S						

Learning with Constraints:
Experiments: Typenet

	MAP Scores			Constraint Violations		
Scenario	**5% Data**	**10% Data**	**100% Data**	**5% Data**	**10% Data**	**100% Data**
B	68.6			22,715		
B+H	68.71			22,928		
B+C	80.13			**25**		
B+S	**82.22**			41		

Learning with Constraints: *Experiments:* Typenet

	MAP Scores			Constraint Violations		
Scenario	**5% Data**	**10% Data**	**100% Data**	**5% Data**	**10% Data**	**100% Data**
B	68.6	69.2	70.5	22,715	21,451	22,359
B+H	68.71	69.31	71.77	22,928	21,157	24,650
B+C	80.13	81.36	**82.80**	**25**	45	**12**
B+S	**82.22**	**83.81**		41	**26**	