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]*(credit: Vivek S Kumar)*
- ➔ **Constraints:** One of the ways of representing symbolic knowledge. $\mathbb{1}\{y_{PER.} = 1\} \implies \mathbb{1}\{y_{Noun.} = 1\}$

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)*
- ➔ **Constraints:** One of the ways of representing symbolic knowledge. $\mathbb{1}\{y_{PER.} = 1\} \implies \mathbb{1}\{y_{Noun.} = 1\}$
- ➔ Limited work in training DL models with (soft) constraints

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)*
- ➔ **Constraints:** One of the ways of representing symbolic knowledge. $\mathbb{1}\{y_{PER.} = 1\} \implies \mathbb{1}\{y_{Noun.} = 1\}$
- ➔ Limited work in training DL models with (soft) constraints
- ➔ What if constraints are hard?

Neural + Constraints

❖ Augmenting deep neural models (**DNN**) with Domain Knowledge (**DK**)

❖ **Domain Knowledge** expressed in the form of ***Constraints*** **(C)**

- **Learning with (hard) constraints:** Learn **DNN** weights s.t. output satisfies **constraints C**

Related Work

		INFERENCE	TRAINING
CONSTRAINTS	SOFT		
	HARD		

Related Work

CONSTRAINTS		INFERENCE	TRAINING
	SOFT	• Gradient based inference (Lee *et al.* ['19]) • Neural+CRF as post processing (Chen *et al* ['18]) **DL**	• Semantic loss (Xu *et al.* ['18]) • Semi-supervised SRL (Mehta *et al.* ['18]) • Posterior Regularization + Distillation (Hu *et al.* ['16]) **DL**
	HARD	• CCM (Roth & Yih [2005], Chang et al [2013]) • Dual Decomposition (Rush & Collins [2012]) **Non DL**	**Our Work** **DL**

Learning with Constraints: *Running Example*

- **Task:** Fine Grained Entity Typing

Learning with Constraints: *Running Example*

Input: Bag of Mentions

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

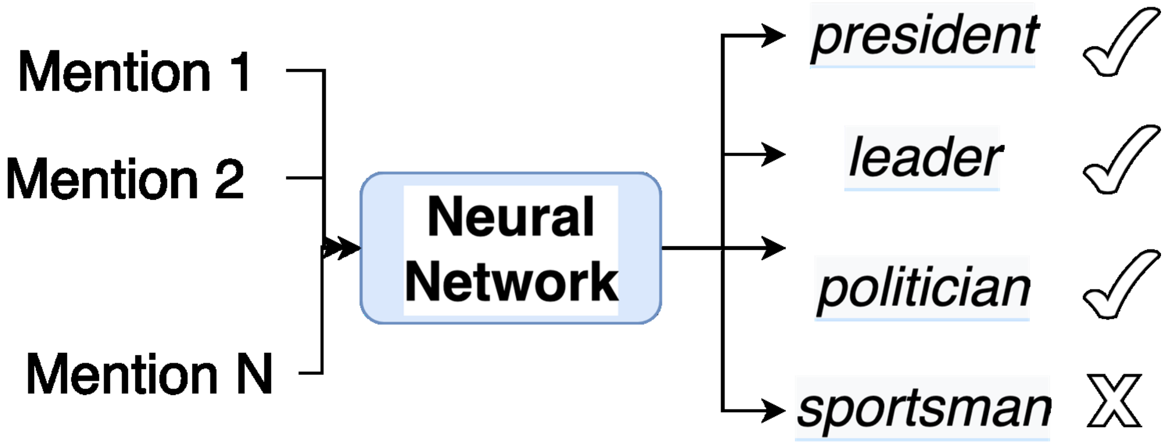
Output:
president, leader, politician...

Learning with Constraints: *Running Example*

Input: Bag of Mentions

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

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

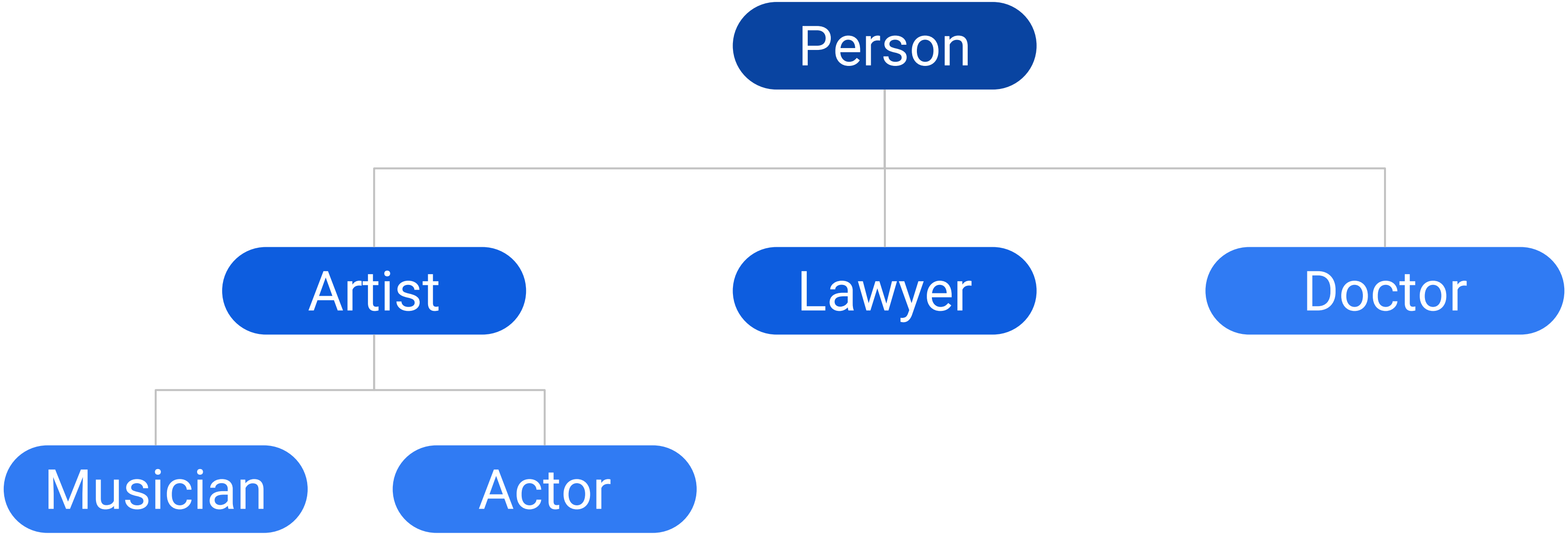


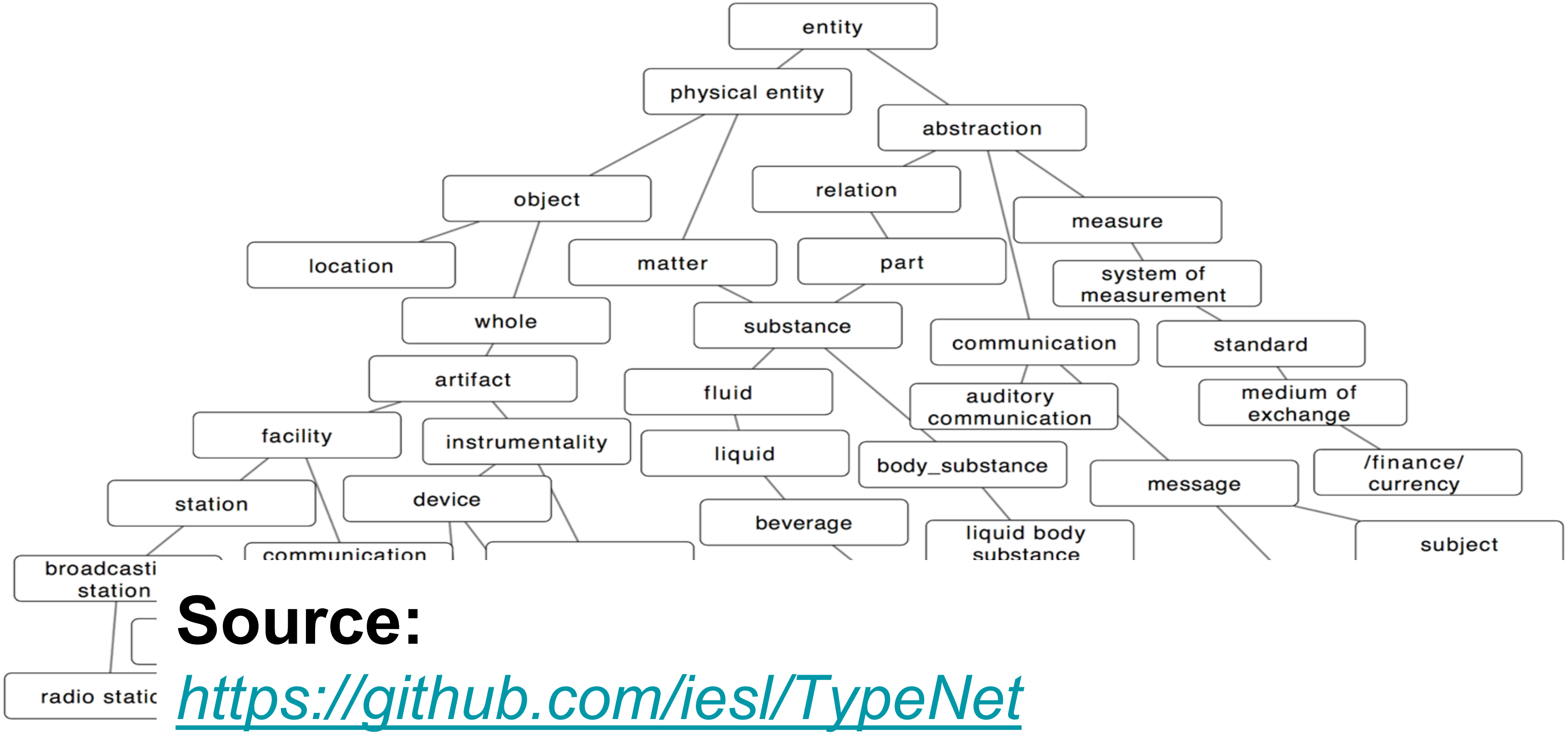
Learning with Constraints: *Running Example*

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

Learning with Constraints: *Running Example*

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





Source:

https://github.com/iesl/TypeNet

https://github.com/MurtyShikhar/Hierarchical-Typing

Learning with Constraints: *Representation of Constraints*

➔ Using Soft Logic

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

Learning with Constraints: *Representation of Constraints*

→ Using Soft Logic

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

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

Learning with Constraints: *Representation of Constraints*

→ Using Soft Logic

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

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

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

Le

Cc

→ l

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

$$\mathbb{1}\{y_{\textsc{Artist}}=1\} \implies \mathbb{1}\{y_{\textsc{Person}}=1\}$$

$$(\neg\mathbb{1}\{y_{\textsc{Artist}}=1\}) \vee (\mathbb{1}\{y_{\textsc{Person}}=1\})$$

$$(1-p(y_{\textsc{Artist}}))+p(y_{\textsc{Person}})$$

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

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

Learning with Constraints: *Formulation*

Constrained Problem

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

Where:

m: Size of training data

K: Number of Constraints

Learning with Constraints: *Formulation*

Constrained Problem

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

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

Learning with Constraints: *Formulation*

Constrained Problem

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

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

$$\min_{w} \max_{\Lambda} \mathcal{L}(w, \Lambda) \qquad \geq \qquad \max_{\Lambda} \min_{w} \mathcal{L}(w, \Lambda)$$

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:

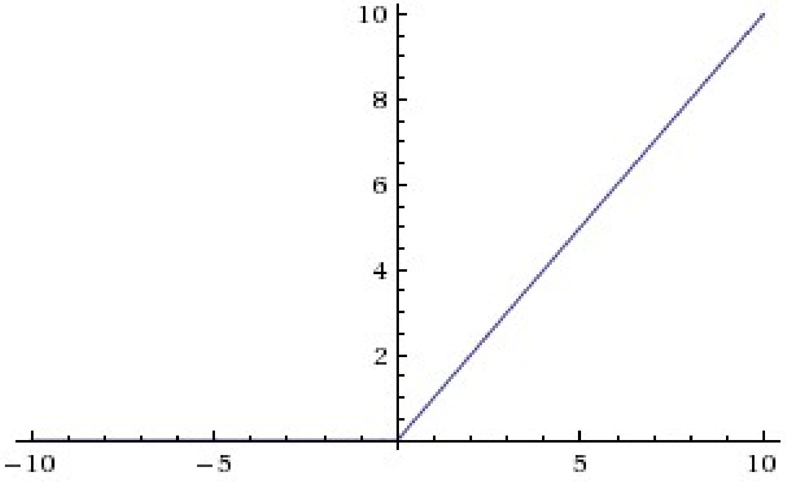
O(mK) #constraints

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

Learning with Constraints: *Reduce # Constraints*

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

H(c)



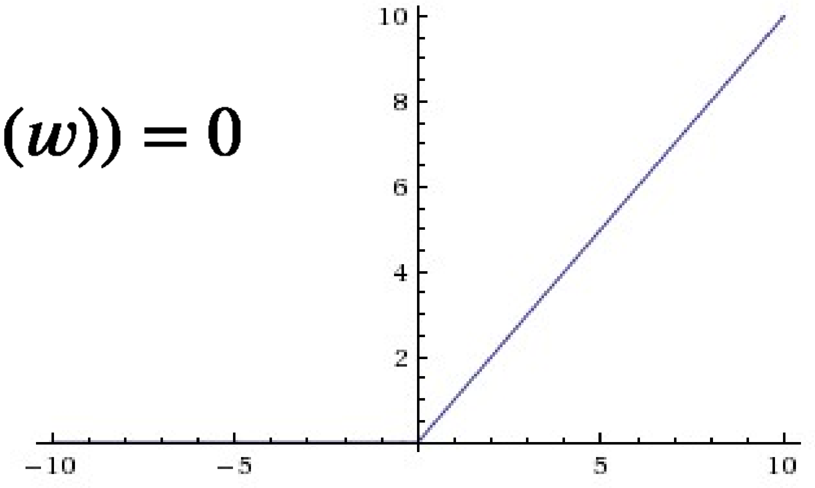
Learning with Constraints: *Reduce # Constraints*

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

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

Equivalent

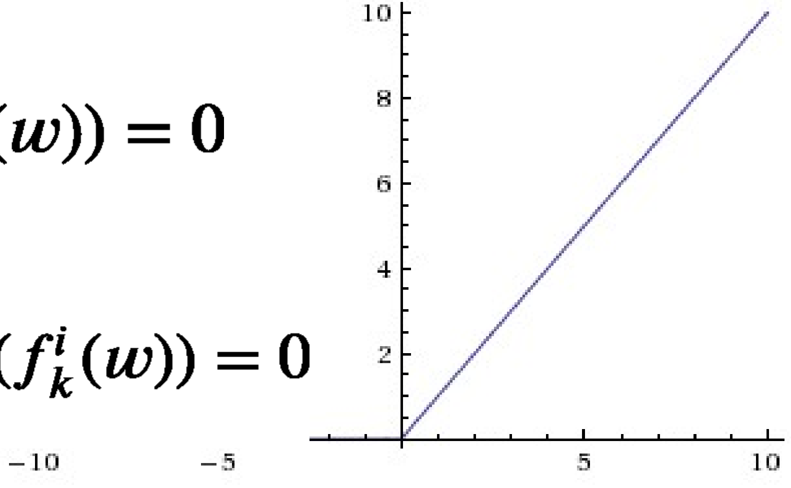
H(c)



Learning with Constraints: *Reduce # Constraints*

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

H(c)



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

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:

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

Now:

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

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

Learning with Constraints: *Reduce # Constraints*

Originally:

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

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: *Primal-Dual Formulation*

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

Lagrangian

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

Learning with Constraints: *Primal-Dual Formulation*

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

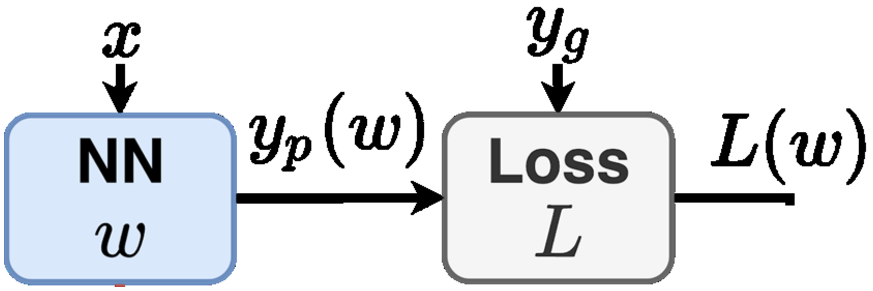
Lagrangian

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

Primal **Dual**

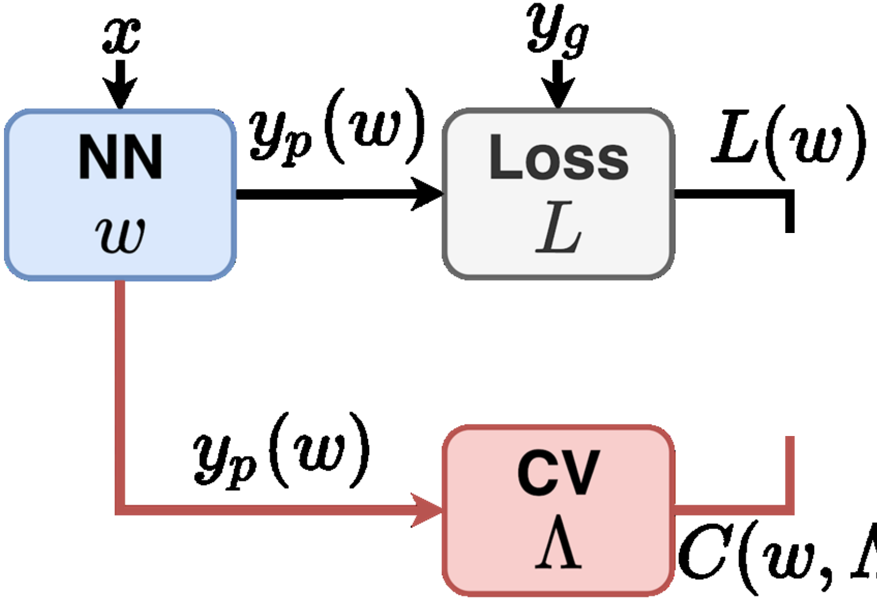
$$\min_{w} \max_{\Lambda} \mathcal{L}(w, \Lambda) \quad \ge \quad \max_{\Lambda} \min_{w} \mathcal{L}(w, \Lambda)$$

Learning with Constraints: *Parameter Update*



w **Update**

Learning with Constraints: *Parameter Update*

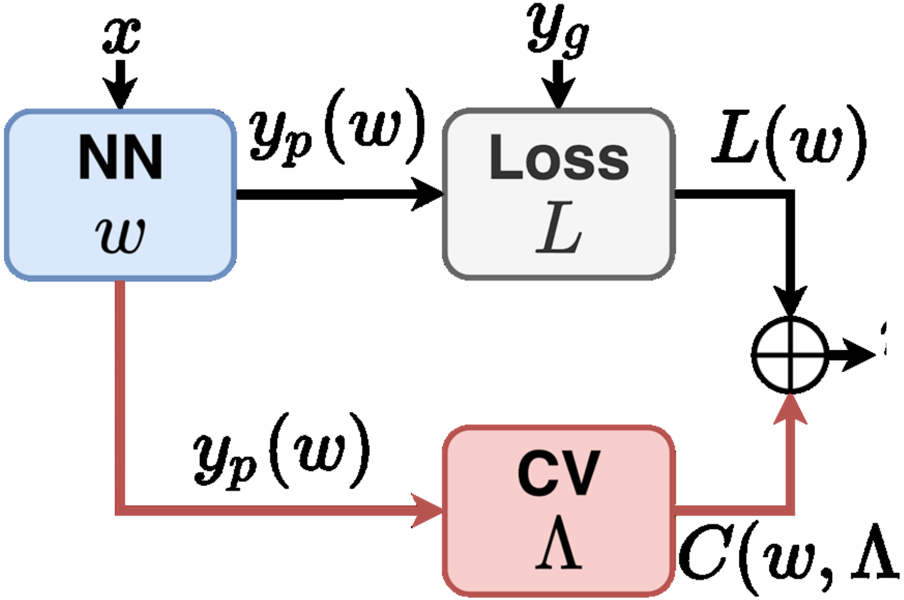


$C(w, \Lambda) = \sum_{k=1}^{K} \lambda_k h_k(w)$

ate

Λ Fixed

Learning with Constraints: *Parameter Update*



$$\mathcal{L}(w, \Lambda) = L(w) + C(w, \Lambda)$$

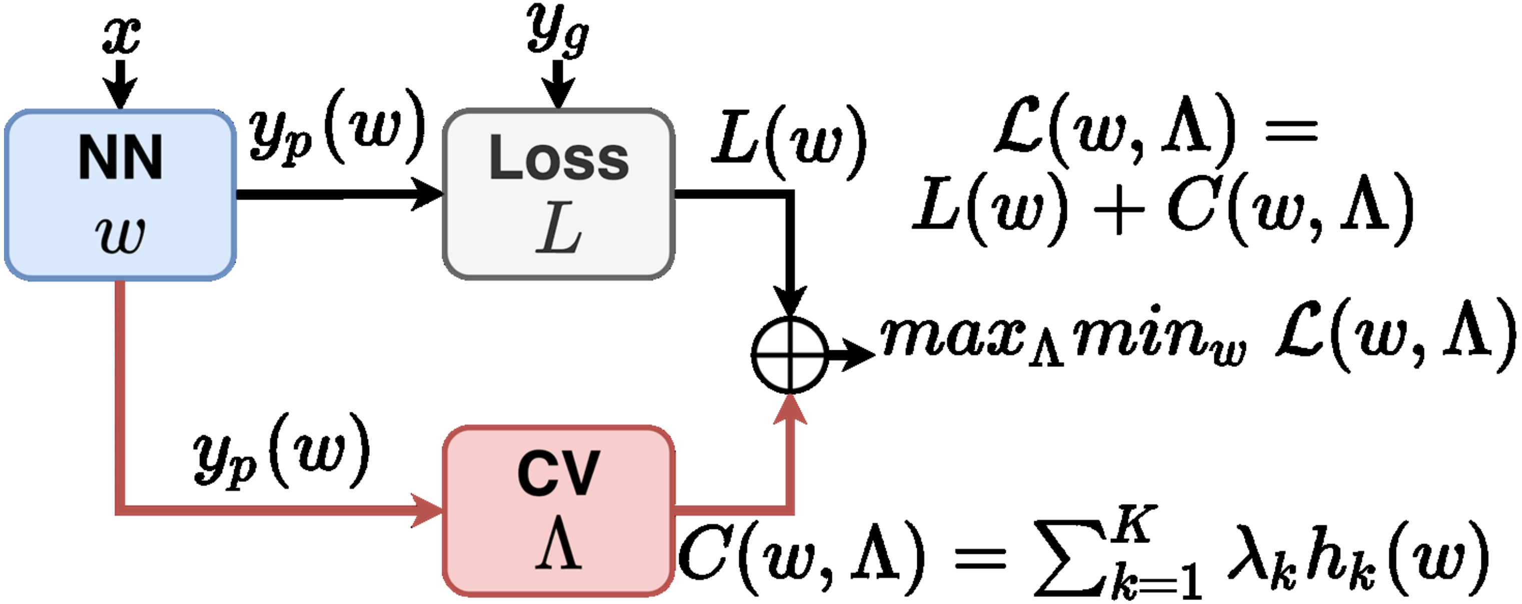
$$min_w \; \mathcal{L}(w, \Lambda)$$

$$C(w, \Lambda) = \sum_{k=1}^{K} \lambda_k h_k(w)$$

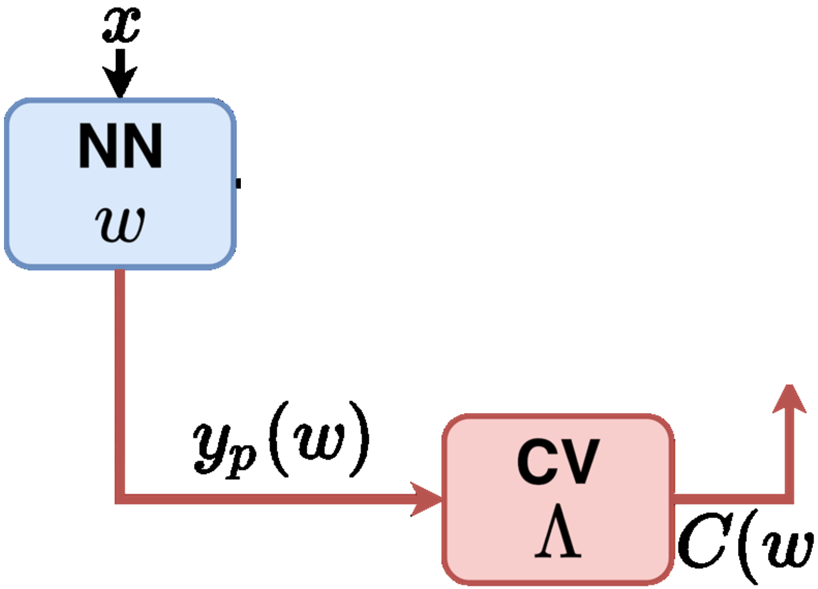
w Update

Λ Fixed

Learning with Constraints: *Parameter Update*



Learning with Constraints: *Parameter Update*

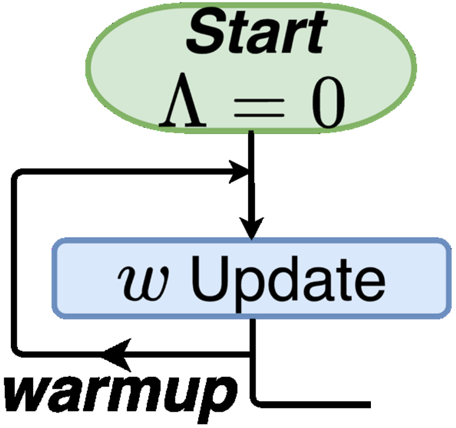


$$C(w, \Lambda) = \sum_{k=1}^{K} \lambda_k h_k(w)$$

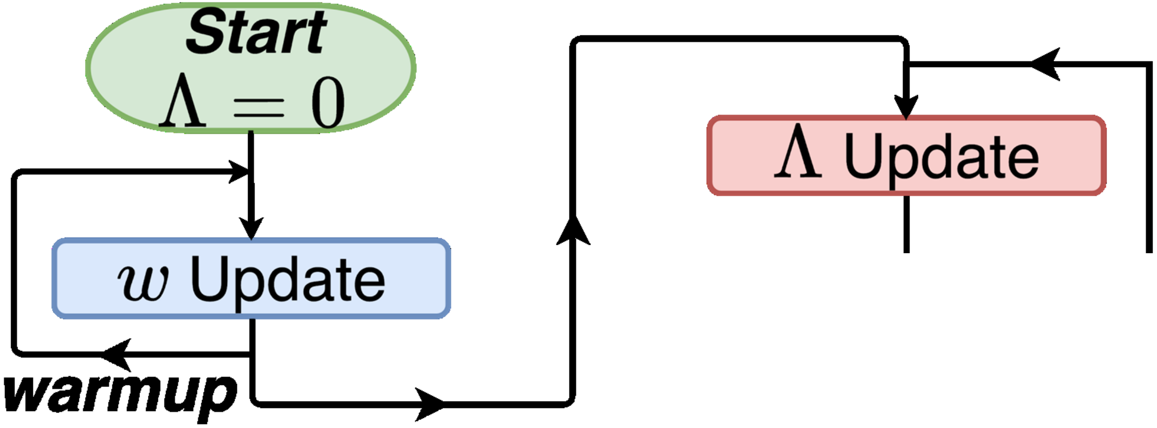
w *Fixed*

Λ *Update*

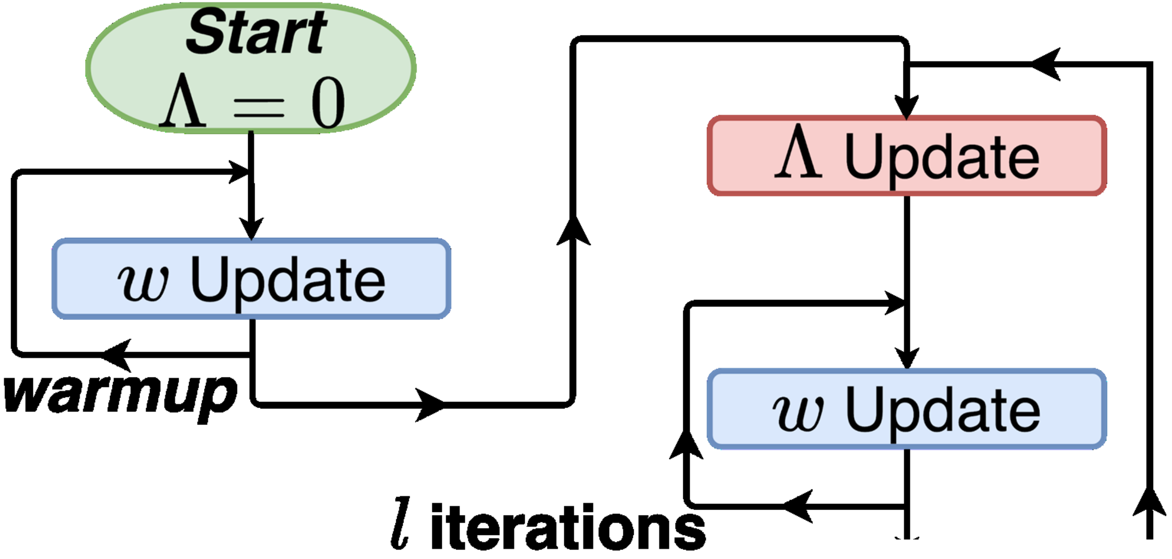
Learning with Constraints: *Training Algorithm*



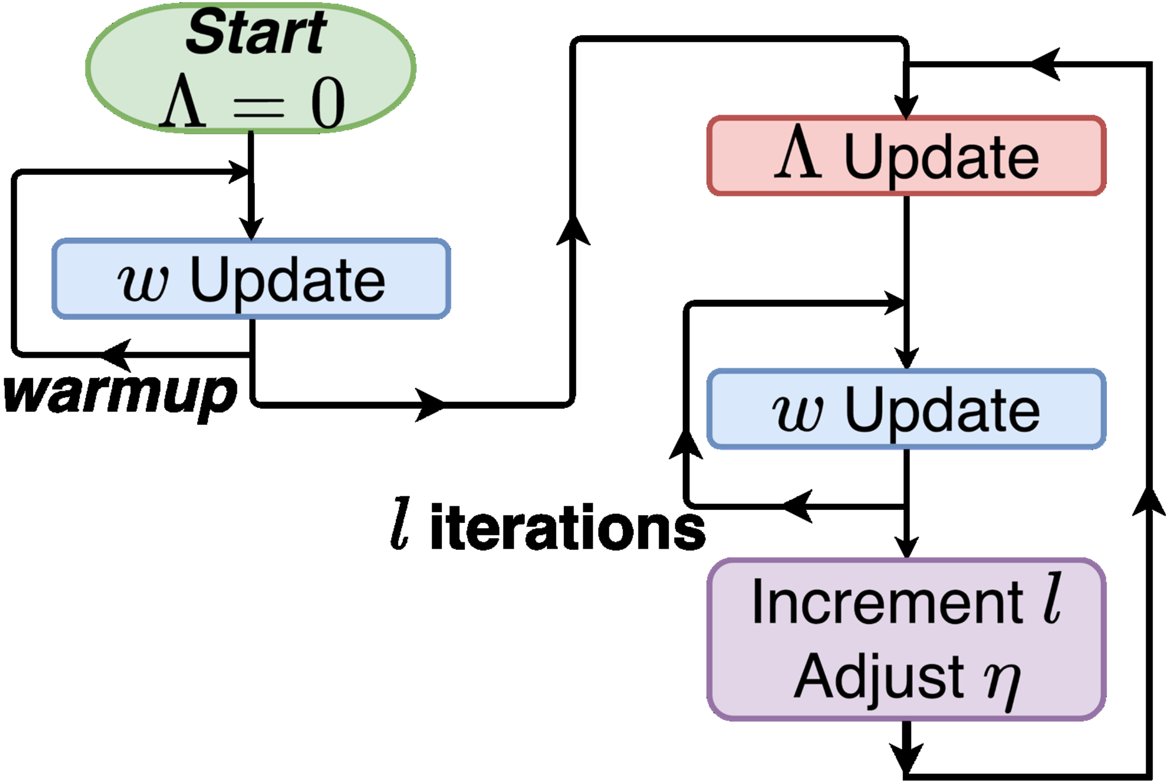
Learning with Constraints: *Training Algorithm*



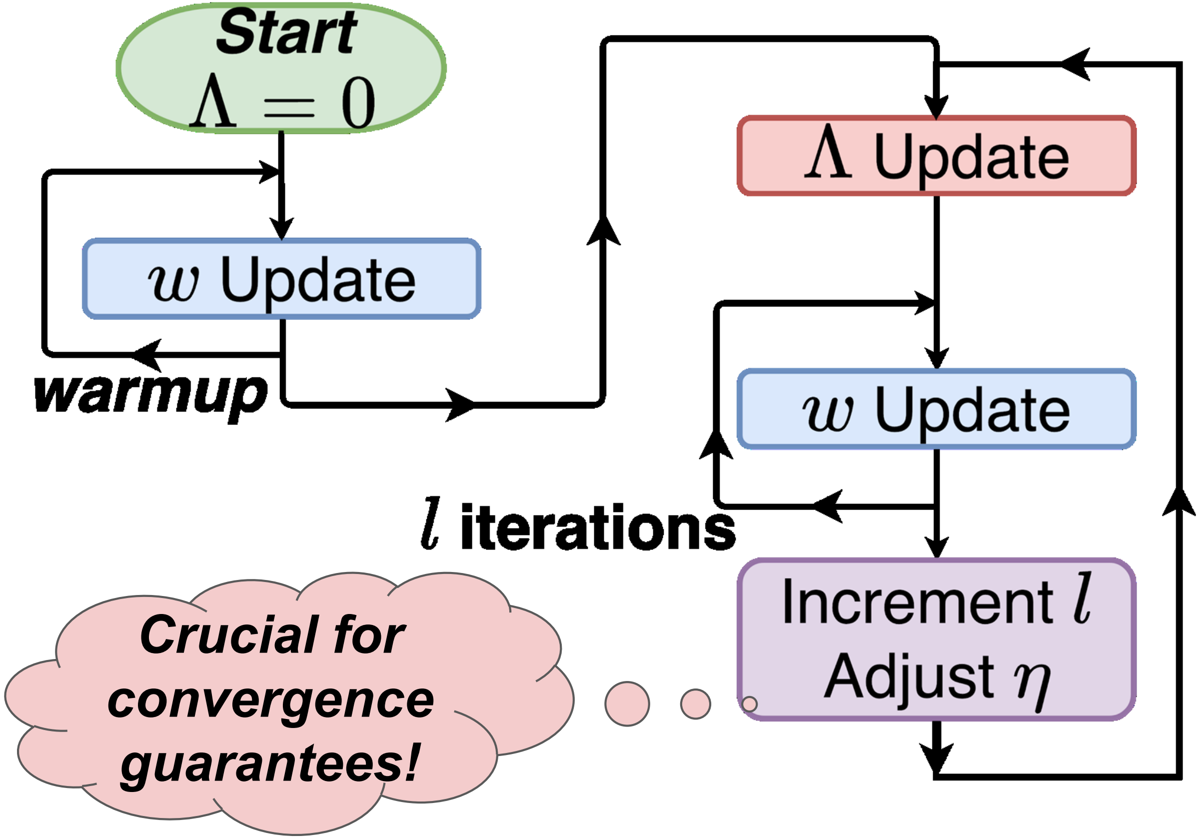
Learning with Constraints: *Training Algorithm*



Learning with Constraints: *Training Algorithm*



Learning with Constraints: *Training Algorithm*



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**	

Learning with Constraints: *Experiments*
NER

Task: Named Entity Recognition

Auxiliary Task: Part of Speech Tagging

Learning with Constraints: *Experiments*
NER

Task: Named Entity Recognition

Auxiliary Task: Part of Speech Tagging

Architecture: Common LSTM encoder and task specific classifier

Learning with Constraints: *Experiments*
NER

Task: Named Entity Recognition

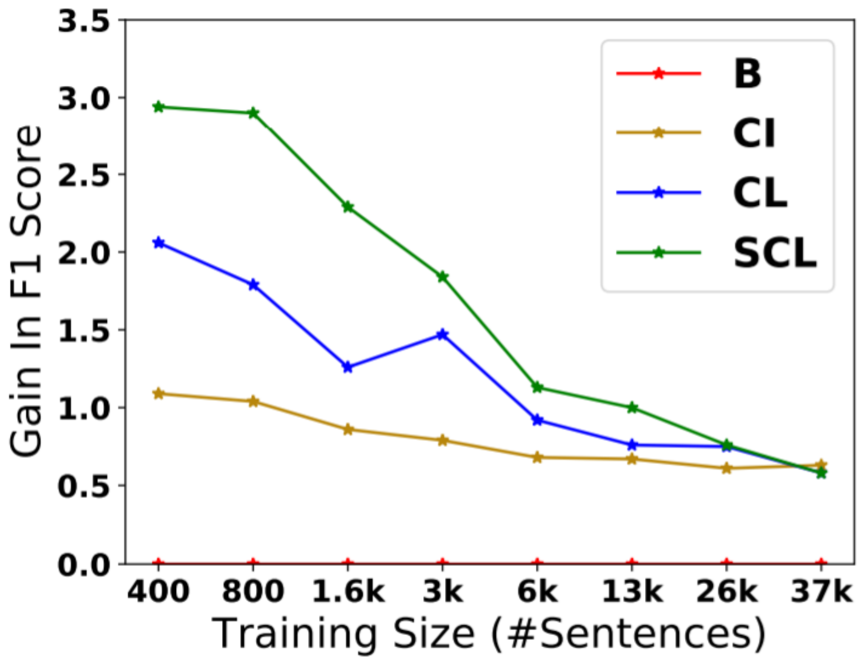
Auxiliary Task: Part of Speech Tagging

Architecture: Common LSTM encoder and task specific classifier

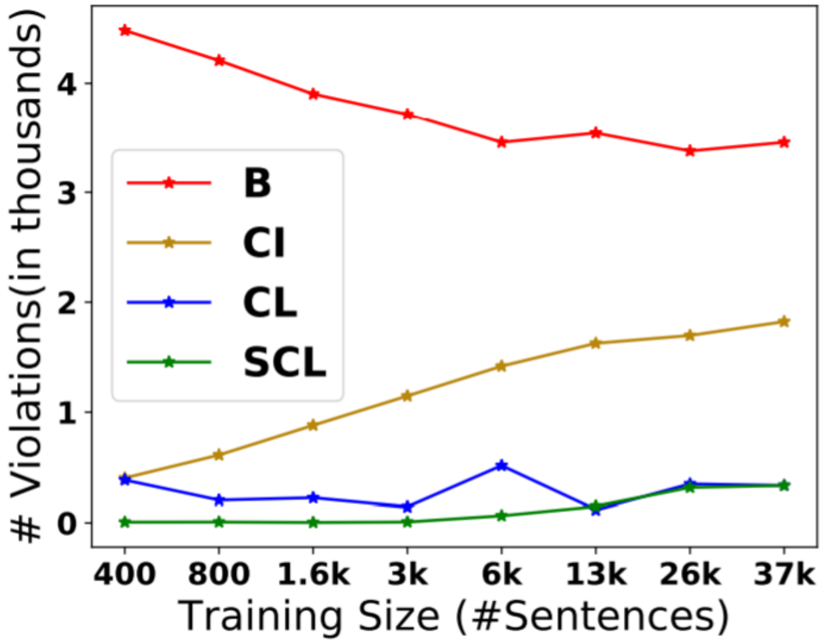
Constraints: 16 constraints of type: *Person => Noun*

Learning with Constraints: *Experiments*

NER



(a) Avg. Gain in F1 Score Over Baseline.



(b) Avg. number of Constrained Violations

Learning with Constraints: *Experiments*
SRL

Task: Semantic Role Labelling

Auxiliary Info: Syntactic Parse Trees

Learning with Constraints: *Experiments*

SRL

- For each clause, determine the semantic role played by each noun phrase that is an argument to the verb.

  agent   patient   source   destination   instrument

  – John drove Mary from Austin to Dallas in his Toyota Prius.

  – The hammer broke the window.

- Also referred to a "case role analysis," "thematic analysis," and "shallow semantic parsing"

Slide Credit: Ray Mooney

Learning with Constraints: *Experiments*

SRL

Task: Semantic Role Labelling

Auxiliary Info: Syntactic Parse Trees

Architecture: State-of-the-art based on ELMo embeddings

Learning with Constraints: *Experiments*
SRL

Task: Semantic Role Labelling

Auxiliary Info: Syntactic Parse Trees

Architecture: State-of-the-art based on ELMo embeddings

Constraints: Transition Constraints & span constraints

Learning with Constraints: *Experiments*

SRL

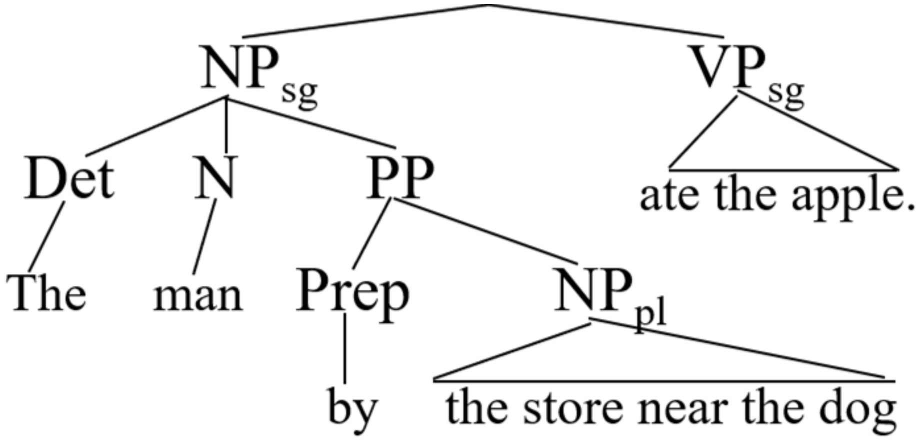
Constraints:

Transition Constraints e.g. **B-Arg(*i*) => I-Arg(*i+1*)**

Span Constraints: Semantic spans should be subset of syntactic spans

Learning with Constraints: *Experiments*

SRL: Syntactic Parse Tree for span constraints



"The man by the store near the dog ate an apple."

"The man" is the agent of "ate" not "the dog".

Slide Credit: Ray Mooney

Learning with Constraints: *Experiments*

SRL

	F1 Score			Total Constraint Violations		
Scenario	**1% Data**	**5% Data**	**10% Data**	**1% Data**	**5% Data**	**10% Data**
B	62.99			14,857		
CL	66.21			9,406		
B+CI						
CL + CI						

Learning with Constraints: *Experiments*

SRL

	F1 Score			Total Constraint Violations		
Scenario	**1% Data**	**5% Data**	**10% Data**	**1% Data**	**5% Data**	**10% Data**
B	62.99	72.64	76.04	14,857	9,708	7,704
CL	66.21	74.27	77.19	9,406	7,461	5,836
B+CI						
CL + CI						

Learning with Constraints: *Experiments*

SRL

	F1 Score			Total Constraint Violations		
Scenario	**1% Data**	**5% Data**	**10% Data**	**1% Data**	**5% Data**	**10% Data**
B	62.99	72.64	76.04	14,857	9,708	7,704
CL	66.21	74.27	77.19	9,406	7,461	5,836
B+CI	67.9	75.96	78.63	5,737	4,247	3,654
CL + CI	68.71	76.51	78.72	5,039	3,963	3,476

Reviews

Doubt

1. Why constraint violations even though they are hard.

Reviews

Weakness

1. Design of constrain function requires significant background knowledge about the task. [Jigyasa]
2. I think we cannot model constraints that are dependent on surrounding generated text. Like a sorting task, with unknown no. of numbers. Generated sequence should have $t_i < t_j$ if $i < j$.

Reviews

Extension

1. **Other Domains:** robotics (physical constraints like reachability, physical properties of objects etc).
2. **Learning Constraints:** Latent representation over the space of logical symbols to fill 3 slots like A --> B. Now, whatever this latent representation is suggesting as a constraint, take that as a hard constraint over the next epoch. This can be extended to have a fixed number of constraints in the model. This would be like learning constraints from the given sample of data, whether that is good or bad, I am not sure because a dataset usually consists of biases in various forms.

References

1. Z. Hu, X. Ma, Z. Liu, E. H. Hovy, and E. P. Xing. Harnessing deep neural networks with logic rules, *ACL 2016*
2. C. Jin, P. Netrapalli, & M. I. Jordan. Minmax optimization: Stable limit points of gradient descent ascent are locally optimal, *arxiv 2019*
3. S. V. Mehta, J. Y. Lee, and J. G. Carbonell. Towards semi-supervised learning for deep semantic role labeling. *AAAI 2019*
4. S. Murty, P. Verga, L. Vilnis, I. Radovanovic, and A. McCallum. Hierarchical losses and new resources for fine-grained entity typing and linking, *ACL 2018*
5. J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. *ICML 2018*

Thank You!