Gradient-based Inference for Networks with Output Constraints

Jay Yoon Lee† Sanket Vaibhav Mehta† Michael Wick‡
Jean-Baptiste Tristan‡ Jaime Carbonell†

‡ ORACLE

† Carnegie Mellon University
Carnegie Mellon
SCHOOL OF COMPUTER SCIENCE

Output constraints?

Gradient-based Inference for Networks

with **Output Constraints**

Constraints?

- **Grammar (example 1)**
  - I are presenting on my thesis proposal.
  - I am presenting on my thesis proposal.

  - The keys is on the table.
  - The keys are on the table.

  - The keys to the cabinet is on the table.
  - The keys to the cabinet are on the table.

Linzen et al, Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies, 2016

Constraints?

- **Grammar (example 1)**
  - I are presenting on my thesis proposal.
  - I am presenting on my thesis proposal.

  - The keys is on the table.
  - The keys are on the table.

  - The keys to the cabinet is on the table.
  - The keys to the cabinet are on the table.

Constraint violating

If subject, verb do not agree. (singular/plural)

Linzen et al, Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies, TACL 2016

Constraints?

- **Image-captioning (example 2)**

Input (image)

Output (caption) A woman is talking on a cell phone.

Rohrbach et al, Object Hallucination in Image Captioning, EMNLP 2018

Constraints?

- **Image-captioning (example 2)**

Input (image)

Constraint violating

If an entity/object not present in the picture appears on the caption.

Output (caption)

A woman is talking on a cell phone.

A woman is talking on a cell phone while sitting on a bench.

A woman is talking to a mouth.

A woman is talking to a pizza.

Rohrbach et al, Object Hallucination in Image Captioning, EMNLP 2018

Constraints?

- **Image-captioning (example 2)**

Input (image)

Constraint violating

If an entity/object not present in the picture appears on the caption.

Output (caption)

A woman is talking on a cell phone.

A woman is talking on a cell phone while sitting on a bench.

A woman is talking to a mouth.

A woman is talking to a pizza.

Probably wrong, but would be another constraint.

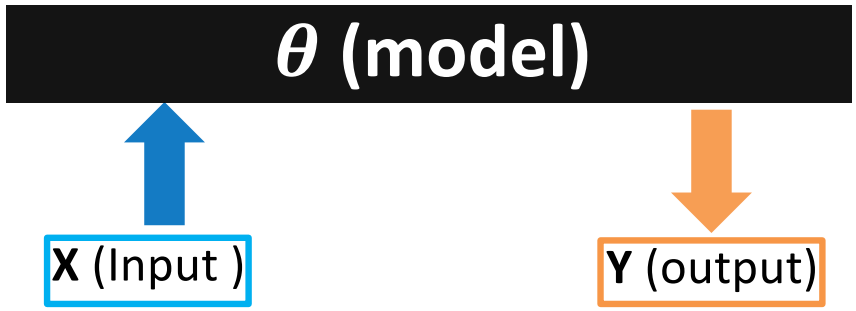
Rohrbach et al, Object Hallucination in Image Captioning, EMNLP 2018

Neural Network

Gradient-based Inference for
Networks with Output Constraints

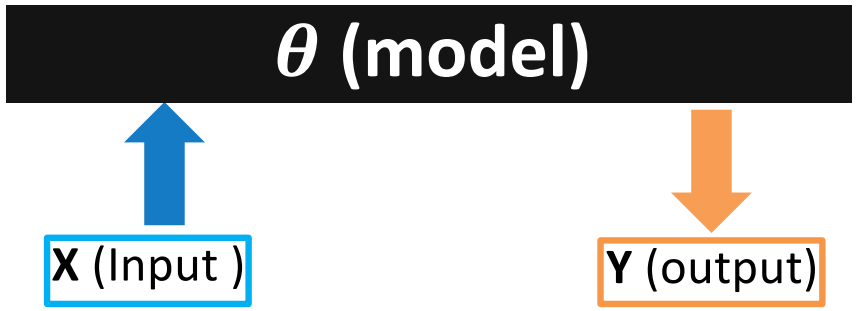
Typical neural model training

- A black box is trained with cross-entropy loss
  - Maximizing log-likelihood of (input, true output) pair.
  - Often requires massive amount of data.



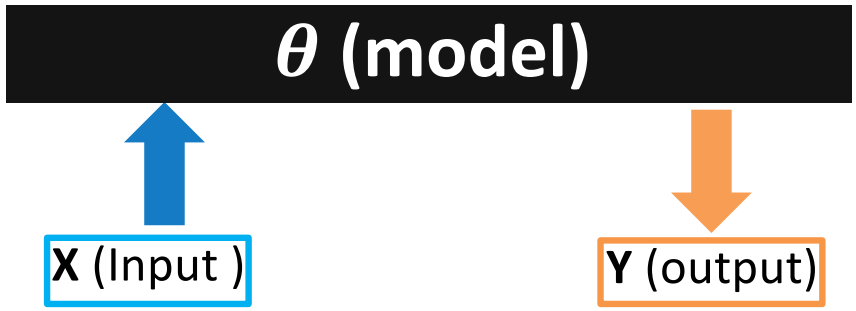
Typical neural model training

- Black box trained with cross-entropy loss
  - Maximizing log-likelihood of (input, true output) pair.
  - Often requires massive dataset. (not very good on small data)
- Can we ***leverage the known constraints?***



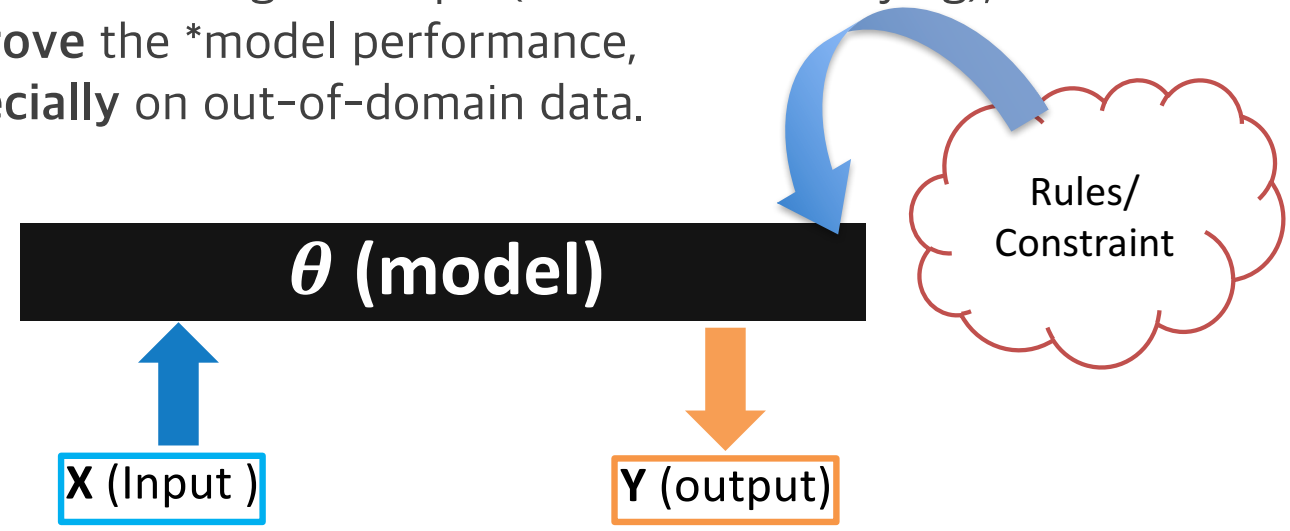
Challenges in injecting constraints

- Training set
  - constraint-satisfying instances (input ,true output)
  - Thus, constraint is ***"not explicit"*** on training set.
- Hard to write 'if-else' inside black box.



Goal of constraint injection

- Injection of constraint to
  - **Provide** more logical output(constraint-satisfying),
  - **Improve** the *model performance,
  - **Especially** on out-of-domain data.



Roadmap

- Focus of our work
- **Applications & constraints**
- Gradient-Based Inference formulation
- Experiments
- Conclusion

Considered Application

- *Transduction (toy)*
  - A transducer (T) is a function from a
    'source language' (L_s) to a 'target language' (L_T).

$$T : \mathcal{L}_S \rightarrow \mathcal{L}_T$$

Transduction constraints

- *Transduction (toy)*
  - A transducer (T) is a function from a 'source language' (L_s) to a 'target language' (L_T).

$$T : \mathcal{L}_S \rightarrow \mathcal{L}_T$$

Multiple (input, output) pairs

[Source] azazbz ⬇ [Target] aaaaaazb

...

[Source] azbzazbzbzaazbzbzbzbzbz ⬇ [Target] aaazbaaazbzbaaaaaazbzbzbzbzb

Transduction constraints

- *Transduction (toy)*
  - A transducer (T) is a function from a 'source language' (L_s) to a 'target language' (L_T).

$$T : \mathcal{L}_S \rightarrow \mathcal{L}_T$$

e.g.)
T: az → aaa
bz → zb

S: azazbzazbzbazbzbzbzbzbz

T: aaaaaazbaaazbzbaaazbzbzbzbzb

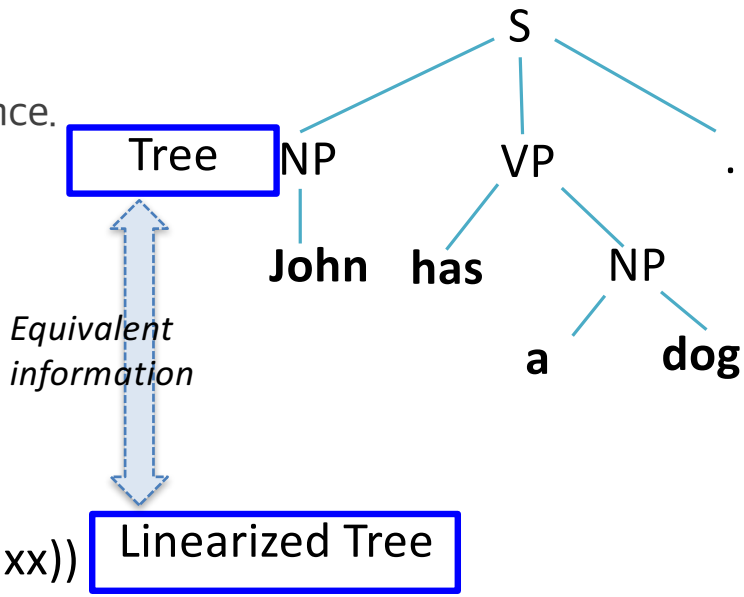
constraint:
3 x (# a in **S**) = # a in **T**

Considered Applications in NLP

- *Syntactic Parsing*
  - Finding structure of the provided (input) sentence.

John has a dog .

(S (NP xx) (VP xx (NP xx xx))

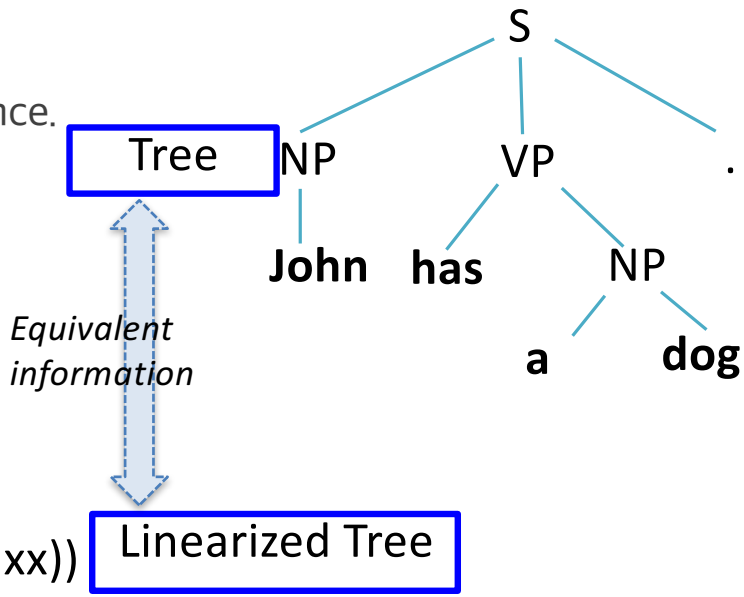


Considered Applications in NLP

- *Syntactic Parsing*
  - Finding structure of the provided (input) sentence.

John has a dog .

(S (NP xx) (VP xx (NP xx xx))



To use *Sequence-to-Sequence* model:

models relationship between *sequence* input & *sequence* output.

Constraints in Parsing

- *Syntactic Parsing*
  - Finding structure of the provided (input) sentence.

John has a dog

True:

(S (NP xx) (VP xx(NP xx xx))

Others:

(S (NP xx) (VP xx (NP xx xx)))))

((S (NP xx) (VP xx (NP xx xx))

(S (NP xx)))(VP xx (NP xx xx)))

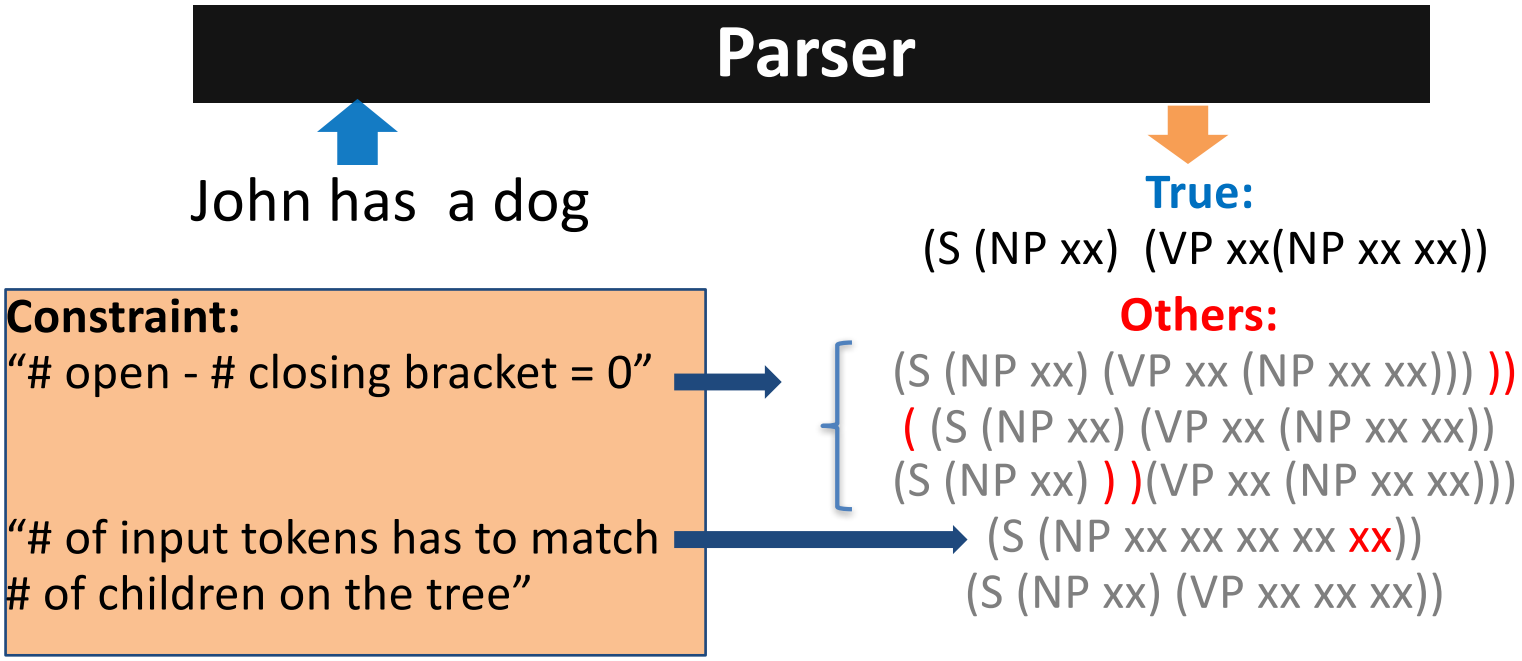
(S (NP xx xx xx xx xx))

(S (NP xx) (VP xx xx xx))

Now you use seq2seq model,
The model might generate constraint-violating outputs

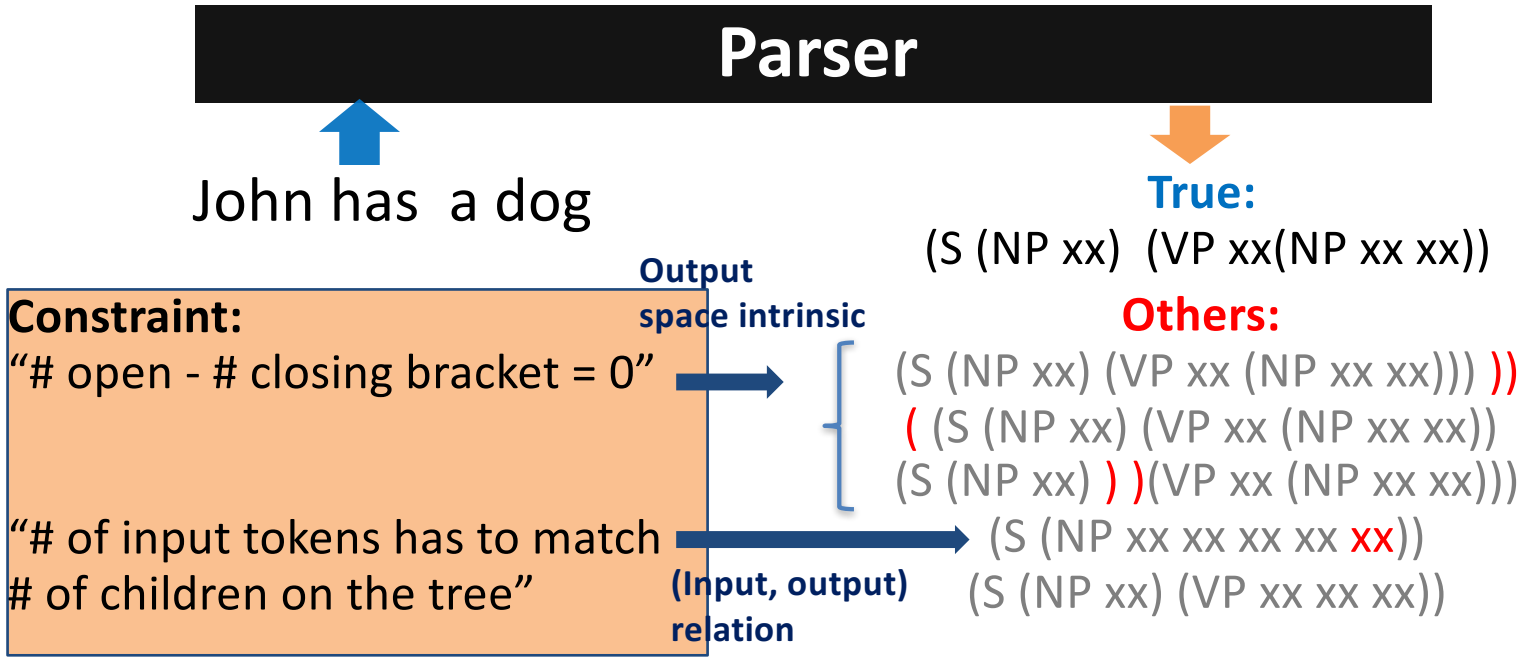
Constraints in Parsing

- *Syntactic Parsing*
  - Finding structure of the provided (input) sentence.



Constraints in Parsing

- *Syntactic Parsing*
  - Finding structure of the provided (input) sentence.



Considered Applications in NLP

- *Semantic Role Labeling (SRL)*
  - Finding (Agent (arg0), Patient (arg1)) given verb

John broke the window with a hammer.
verb: broke

ARG0: John
ARG1: the window
ARG2: a hammer

Semantic Role Labeling

- Application 1: Semantic Role Labeling (SRL)
  - Finding (Agent (arg0), Patient (arg1)) given verb

John broke the window with a hammer.

agent
(actor of an action)

patient
(entity affected by the action)

instrument
(tool used in performing action)

Semantic Role Labeling

- Application 1: Semantic Role Labeling
  - Finding (Agent (arg0), Patient (arg1)) given verb

John broke the window with a hammer.

agent
(actor of an action)

patient
(entity affected by
the action)

instrument
(tool used in
performing action)

Core Roles: Verb specific roles (ARG0, ARG1, …, ARG5)

Verb: break

Role	Description
ARG0	breaker (agent)
ARG1	thing broken (patient)
ARG2	instrument used
…	…

Adjunct roles: Shared across all verbs

Role	Description
ARGM-TMP	temporal
ARGM-DIR	direction
ARGM-LOC	location
…	…

Application using SRL

- **Question answering (5W1H)**
  - "Who" questions usually use Agents. (Arg0)
  - "What" questions usually use Patients. (Arg1)
  - "How" and "with what" questions usually use Instruments. (Arg2)
  - "Where" questions frequently use Sources and Destinations . (ARGM-LOC)
  - "For whom" questions usually use Benefactives. (Arg1)
  - "To whom" questions usually use Destinations.

Semantic Role Labeling

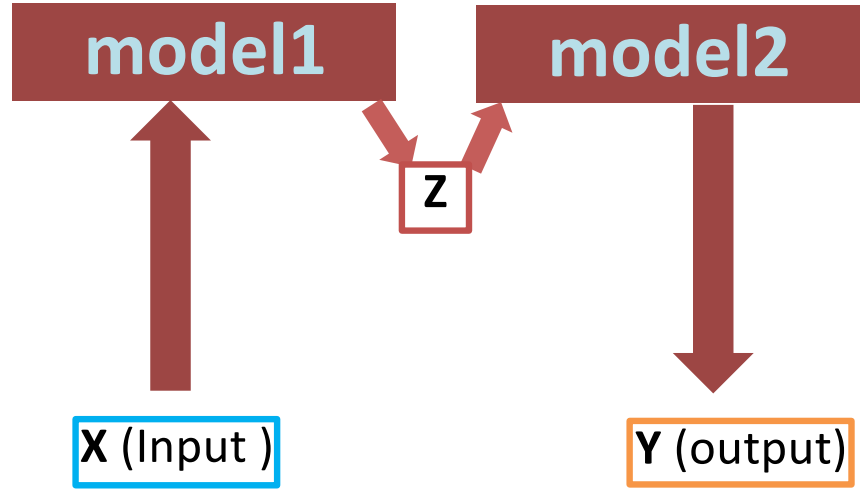
- Application 1: Semantic Role Labeling
  - Finding (Agent (arg0), Patient (arg1)) given verb

John broke the window with a hammer.
verb: broke

ARG0: John
ARG1: the window
ARG2: a hammer

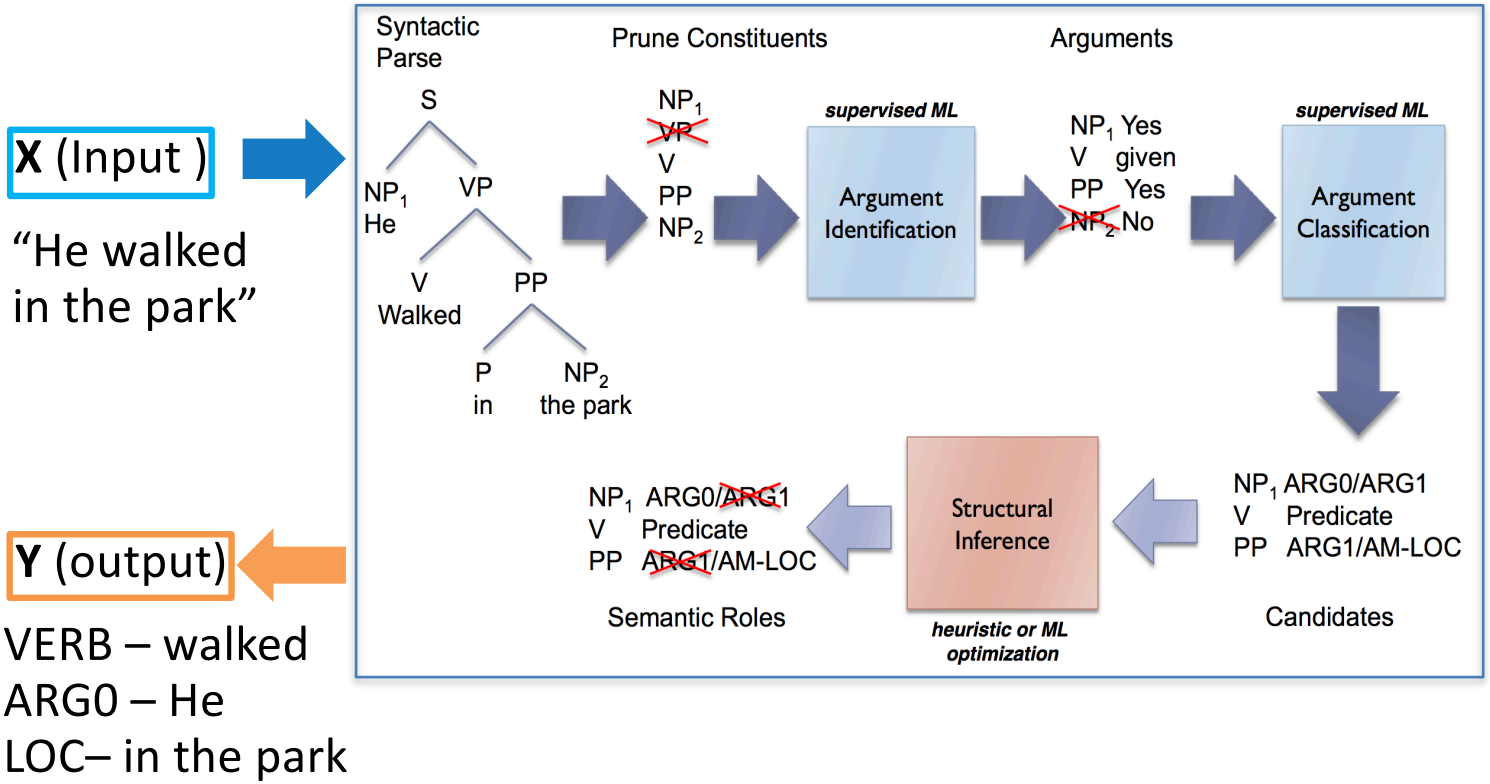
Complex task

- Complex task (sometimes) is sometimes easier to model when the task is divided into multiple ones



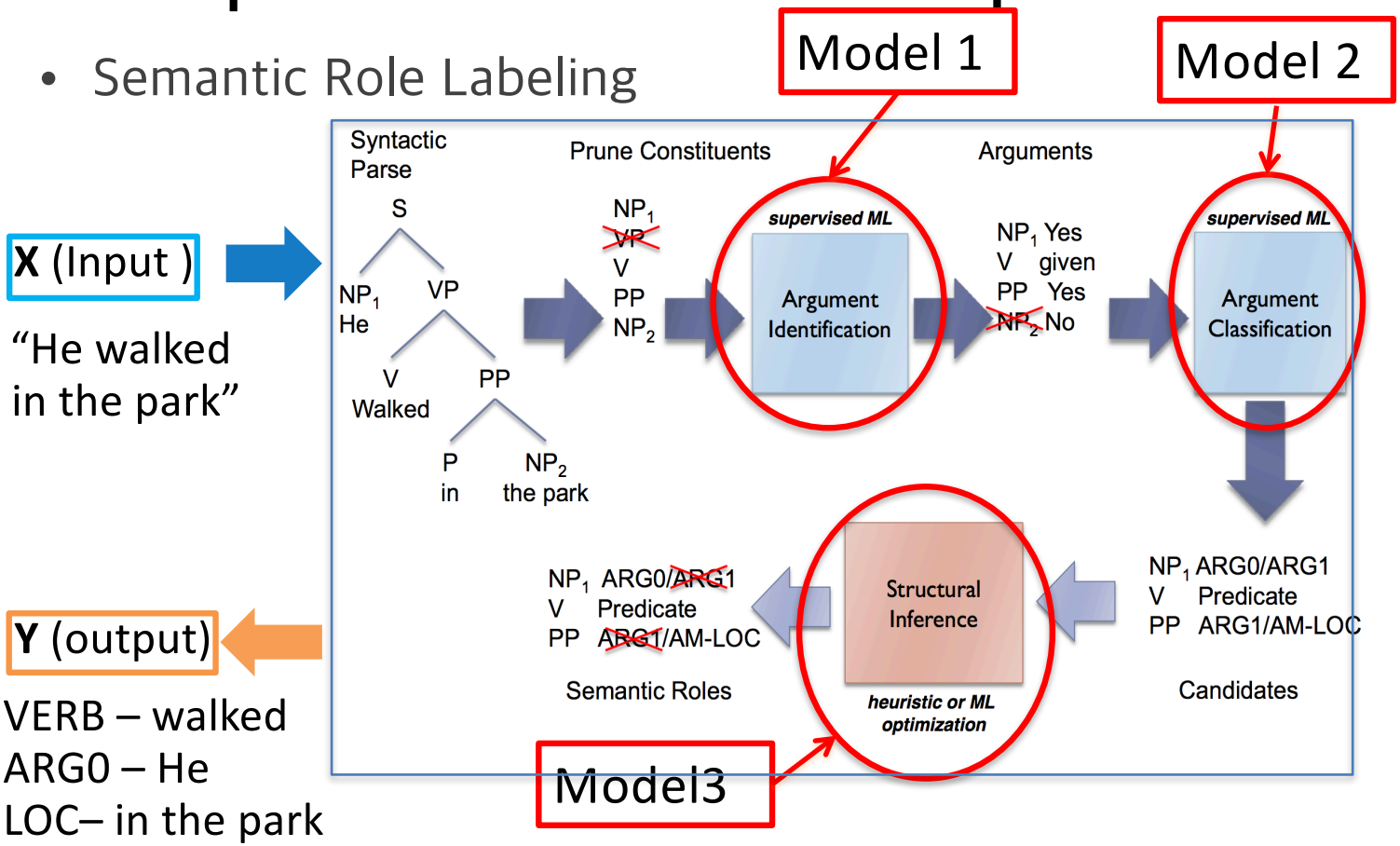
Complex models - example

- Semantic Role Labeling (step by step)



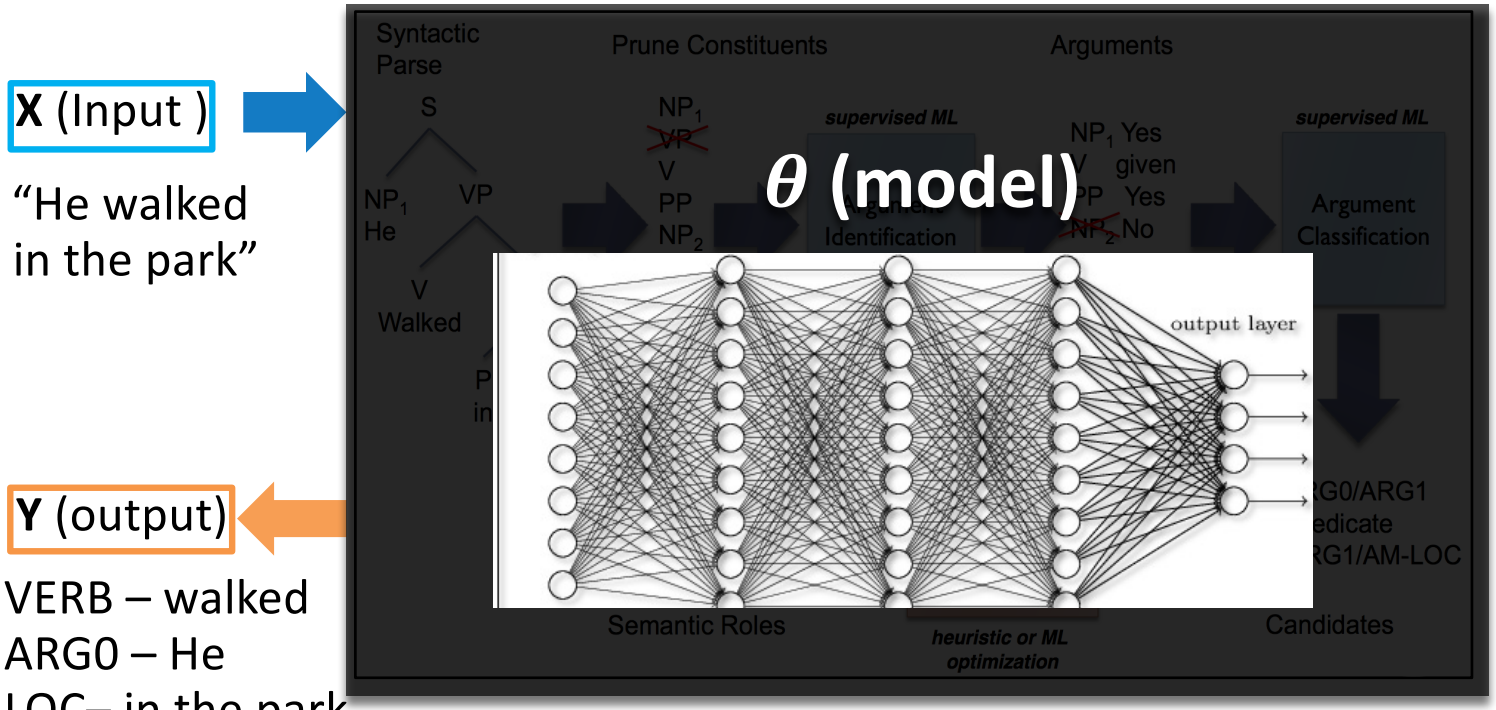
Complex models - example

- Semantic Role Labeling



Complex models to End-to-End

- End-to-end Semantic Role Labeling.
  - Neural models often perform better on ***end-to-end*** learning.



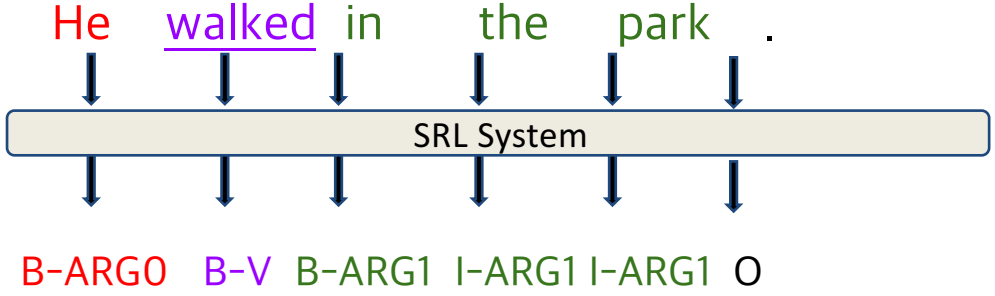
How to solve it?

Formulate the problem as a **BIO tagging problem.**

Input: a sentence-predicate pair (x,verb)

Output: a sequence of tags y, where $y_i \in \Omega$ (a set of BIO tags)

Example:



Slide credits: Sanket Metha

What are different constraints for SRL task?

1. BIO constraints
   - Constraint violation: B-ARG0 followed by I-ARG1
   - (similar to the constraints in Parsing example)

2. SRL constraints (as defined by Punyakanok et al. (2008))
   - Unique core roles (U)
   - Continuation roles (C)
   - Reference roles (R)

3. Syntactic constraints
   - Constraint violation: Arguments that are not constituents within a given parse tree

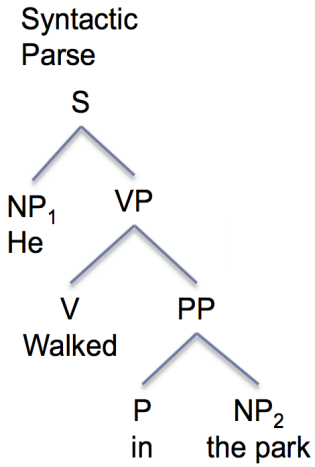
Slide credits: Sanket Metha

Syntactic constraint on SRL

- Given syntax tree, span candidates were ***limited*** by syntactic information.

X (Input)

"He walked in the park"



Possible span candidate

(He), (walked), (in the park),
(walked in the park),
(He walked in the park)

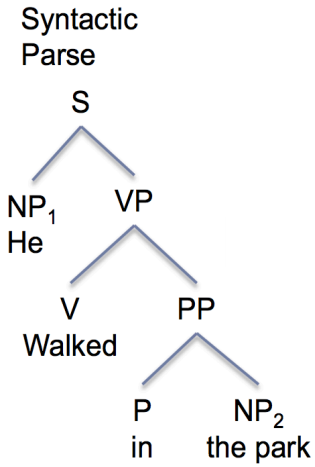
(He) (walked) (in the park)

Syntactic constraint on SRL

- Given syntax tree, span candidates were ***limited*** by syntactic information.

X (Input)

"He walked in the park"



Possible span candidate

(He), (walked), (in the park),
(walked in the park),
(He walked in the park)

(He) (walked) (in the park)
(He) (walked in)(the park)
(He walked) (in the park)
(He walked in the)(park)
(He walked in the park)

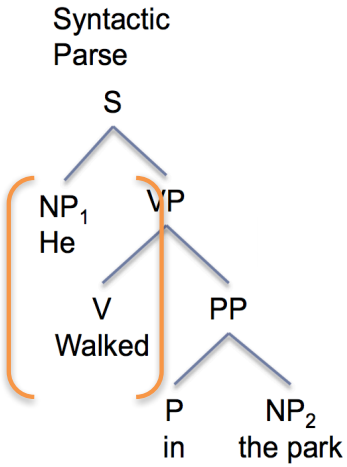
However, end-to-end model without syntactic information can generate **all possible** "spans" given sentence.

Syntactic constraint on SRL

- Given syntax tree, span candidates were ***limited*** by syntactic information.

X (Input)

"He walked in the park"



Possible span candidate

(He), (walked), (in the park),
(walked in the park),
(He walked in the park)

(He) (walked) (in the park)
(He) (walked in)(the park)
~~(He walked)~~ ~~(in the park)~~
(He walked in the)(park)
(He walked in the park)

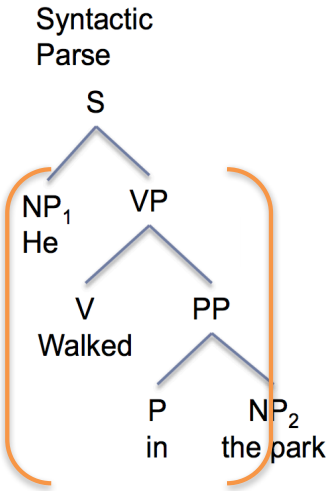
However, end-to-end model without syntactic information can generate **all possible** "spans" given sentence.

Syntactic constraint on SRL

- Given syntax tree, span candidates were ***limited*** by syntactic information.

X (Input)

"He walked in the park"



Possible span candidate

(He), (walked), (in the park),
(walked in the park),
(He walked in the park)

(He) (walked) (in the park)

~~(He)~~ ~~(walked in)~~~~(the park)~~

~~(He walked)~~ ~~(in the park)~~

~~(He walked in the)(park)~~

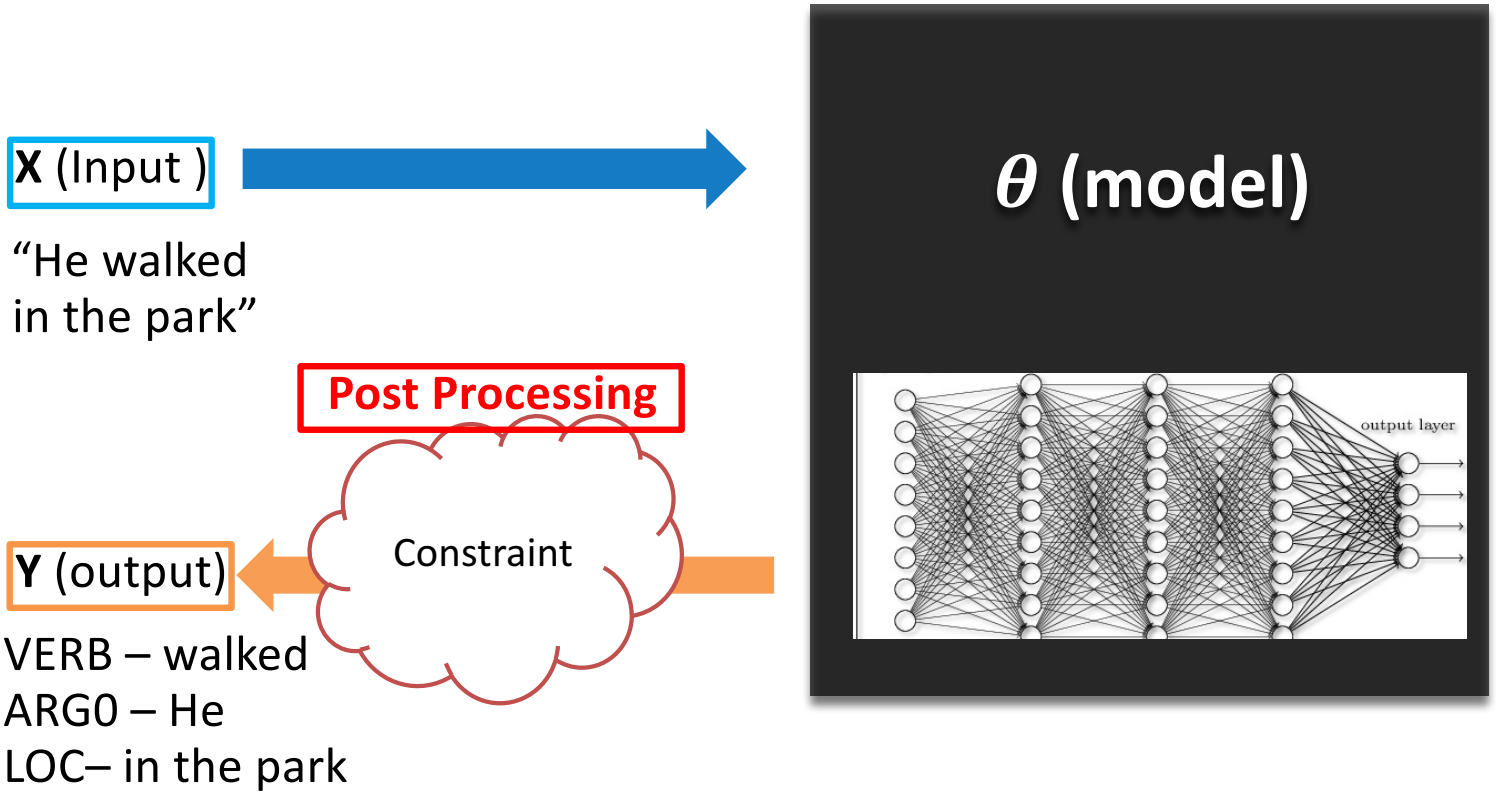
(He walked in the park)

However, end-to-end model without syntactic information can generate all possible "spans" given sentence.

Constraint-violating

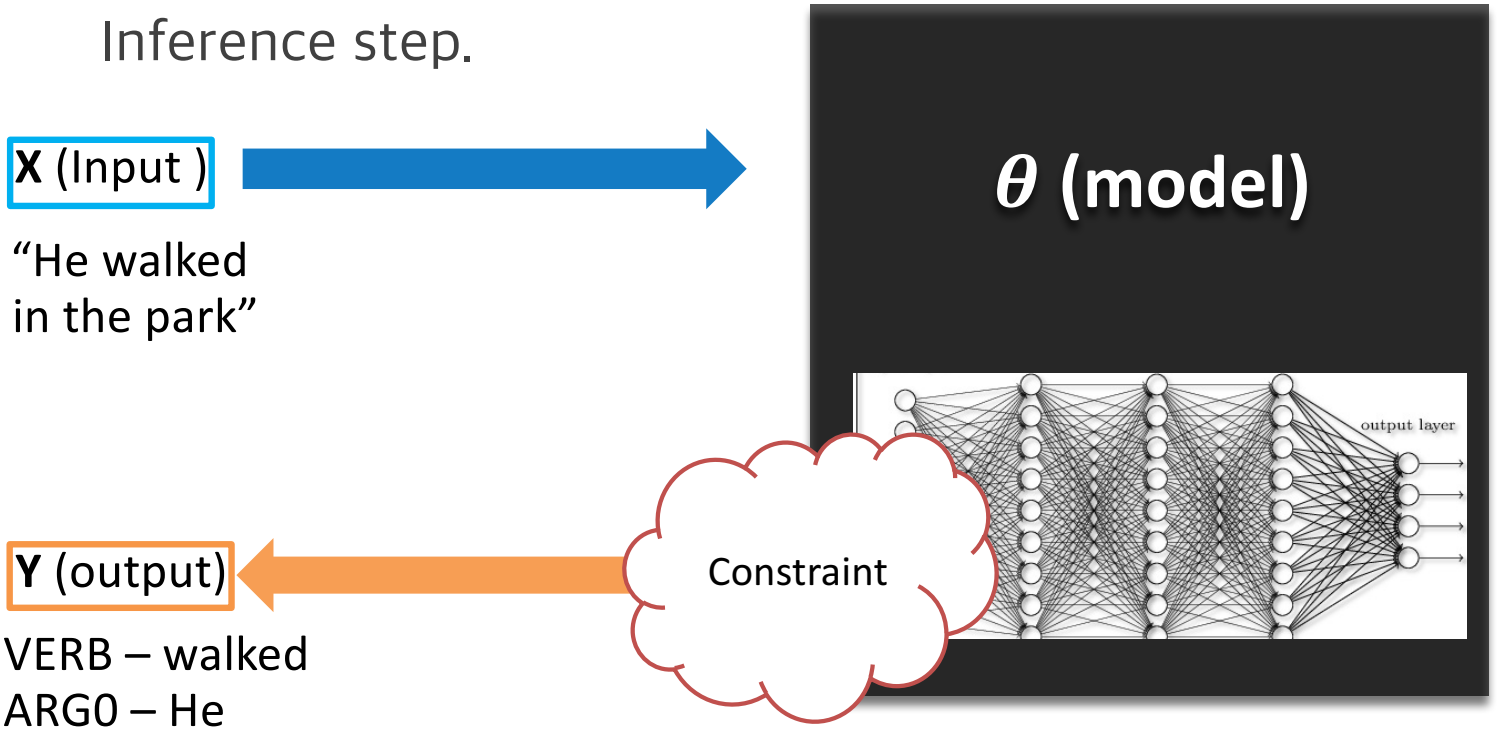
Inference using constraint

- Inference: Finding the best output given the model.



Inference using constraint

- **Proper "Injection of constraints"** to the **model** on Inference step.



Roadmap

- Focus of our work
- Applications & constraints
- **Gradient-Based Inference formulation**
- Experiments
- Conclusion

Gradient-Based Inference* (GBI)

- Inference
  - Finding the output with the highest model probability.

$$\max_{\mathbf{y}} \quad \Psi(\mathbf{x}, \mathbf{y}, W)$$

Learned model

Gradient-Based Inference* (GBI)

- Inference
  - Finding the output with the highest model probability.

$$\max_{\mathbf{y}} \quad \Psi(\mathbf{x}, \mathbf{y}, W)$$

  - Given length *m*, vocabulary size *V,* the search space becomes V^m.
    - Viterbi decoding (exact)
  - Sequence-to-sequence models have no limit to length *m,* thus the search space is larger.
    - Greedy decoding, beam decoding

Constraint function

- Constraint (evaluation) function. (Error counting) $g(\mathbf{y}, \mathcal{L}) \rightarrow \mathbb{R}_0^+$
- no constraint violation $\Longleftrightarrow$ $g(\mathbf{y}, \mathcal{L}) = 0$

Constraint function

- Constraint evaluation function. $g(\mathbf{y}, \mathcal{L}) \rightarrow \mathbb{R}_0^+$
  (degree of error, higher the more error)
- no constraint violation $\longleftrightarrow$ $g(\mathbf{y}, \mathcal{L}) = 0$
- Optimization with constraints

$$\max_{\mathbf{y}} \quad \Psi(\mathbf{x}, \mathbf{y}, W) \qquad \text{s.t.} \quad \mathbf{y} \in \mathcal{L}^{\mathbf{x}}$$

$$\Rightarrow$$

$$\max_{\mathbf{y}} \quad \Psi(\mathbf{x}, \mathbf{y}, W) \qquad \text{s.t.} \quad g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0$$

Constraint
for feasible set L^x

Gradient-Based Inference* (GBI)

- Inference with constraint
  - Finding the output with the highest model probability given (local, global)constraint.

$$\max_{\mathbf{y}} \quad \Psi(\mathbf{x}, \mathbf{y}, W)$$
$$\text{s.t.} \quad g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0$$

constraint

  - Given length *m*, vocabulary size *V*, the search space becomes V^m.
    - Viterbi decoding (exact), **A* algorithm (DP)**
  - Sequence-to-sequence models have no limit to length *m*, thus the search space is larger.
    - Greedy decoding, beam decoding

Gradient-Based Inference* (GBI)

- Inference with constraint
  - Finding the output with the highest model probability given (local, global)constraint.

$$\begin{aligned} \max_{\mathbf{y}} \quad & \Psi(\mathbf{x}, \mathbf{y}, W) \\ \text{s. t.} \quad & g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0 \end{aligned}$$

  - Given length *m*, vocabulary size *V,* the search space becomes V^m.
    - Viterbi decoding (exact), A* algorithm (inexact DP)
  - Sequence-to-sequence models have no limit to length *m,* thus the search space is larger.
    - Greedy decoding, beam decoding

Gradient-Based Inference* (GBI)

- Inference with constraint
  - Finding the output with the highest model probability given (local, global)constraint.

$$\begin{aligned} \max_{\mathbf{y}} \quad & \Psi(\mathbf{x}, \mathbf{y}, W) \\ \text{s. t.} \quad & g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0 \end{aligned}$$

  - Given length *m*, vocabulary size *V,* the search space becomes V^m.
    - Viterbi decoding (exact), A* algorithm (inexact DP)
  - Sequence-to-sequence models have no limit to length *m*, thus the search space is larger.
    - Approximate by greedy decoding, beam decoding

Very slow on exponential search space

Fast but localized approach thus cant incoporate 'global' constraint. (not localizeable)

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\begin{aligned} \max_{\mathbf{y}} \quad & \Psi(\mathbf{x}, \mathbf{y}, W) \\ \text{s.t.} \quad & g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0 \end{aligned}$$

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\begin{aligned} \max_{\mathbf{y}} \quad & \Psi(\mathbf{x}, \mathbf{y}, W) \\ \text{s.t.} \quad & g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0 \end{aligned}$$

Combinatorial problem.

- Gradient-Based Inference (in math)

Can we search faster by changing model weights?

$$\begin{aligned} \min_{W_\lambda} \quad & \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2 \\ \text{where} \quad & \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) \end{aligned}$$

Motivation

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\begin{aligned} \max_{\mathbf{y}} \quad & \Psi(\mathbf{x}, \mathbf{y}, W) \\ \text{s.t.} \quad & g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0 \end{aligned}$$

- Gradient-Based Inference

Likelihood

Constraint violation

$$\begin{aligned} \min_{W_\lambda} \quad & \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2 \\ \text{where} \quad & \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) \end{aligned}$$

Motivation

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\max_{\mathbf{y}} \quad \Psi(\mathbf{x}, \mathbf{y}, W)$$
$$\text{s.t.} \quad g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0$$

Learned model weight

- Gradient-Based Inference

$$\min_{W_\lambda} \quad \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$
$$\text{where} \quad \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$

Expected constraint violation

Regularizer

Motivation

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\max_{\mathbf{y}} \quad \Psi(\mathbf{x}, \mathbf{y}, W)$$
$$\text{s.t.} \quad g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0$$

Standard Lagrangian

$$\min_{\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \lambda g(\mathbf{y}, \mathcal{L})$$

Very slow. (combinatorial)
Especially global constraint

- Gradient-Based Inference (in math)

$$\min_{W_\lambda} \quad \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$
$$\text{where} \quad \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$

Greedy, beam-search
relatively fast

Motivation

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\max_{\mathbf{y}} \quad \Psi(\mathbf{x}, \mathbf{y}, W)$$
$$\text{s.t.} \quad g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0$$

Standard Lagrangian

$$\min_{\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \lambda g(\mathbf{y}, \mathcal{L})$$

Combinatorial problem.

Very slow. (combinatorial)
Especially global constraint

- Gradient-Based Inference (in math)

Now, with expected constraint violation,
we can "reflect" global constraints.

Using the two, can we search faster by changing model weights?

$$\min_{W_\lambda} \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$
$$\text{where} \quad \hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$

Greedy, beam-search
relatively fast

Roadmap

- Focus of our work
- Applications & constraints
- Gradient-Based Inference formulation
- **Research questions & Experiments**
- Conclusion

Research Questions

- **Q1.** Does GBI actually enforce constraints?
- **Q2.** Does GBI enforce constraints
  without compromising the quality of the system?
- **Q3.** Can GBI work over multiple baseline model?
- **Q4.** Performance of GBI on out-of-domain dataset?

Research Questions

- **Q1.** Does GBI actually ***enforce constraints*?**
- **Q2.** Does GBI enforce constraints
  without compromising the F1 score of the system?
- **Q3.** Can GBI generalize over multiple baseline model?
- **Q4.** Performance of GBI on out-of-domain dataset?

Research Questions

- **Q1.** Does GBI actually ***enforce constraints*?**
- **Q2.** Does **GBI enforce constraints**
  without compromising the ***F1 score*** of the system?
- **Q3.** Can GBI generalize over multiple baseline model?
- **Q4.** Performance of GBI on out-of-domain dataset?

Research Questions

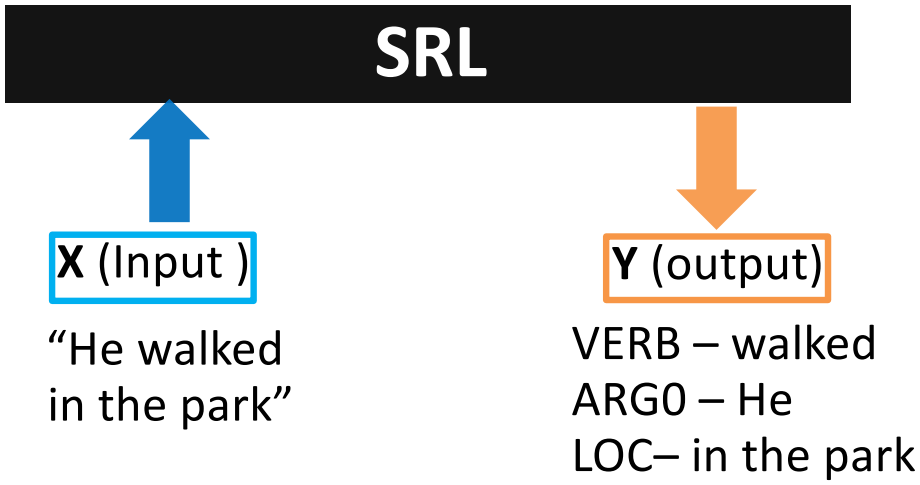
- **Q1.** Does GBI actually ***enforce constraints*****?**
- **Q2.** Does **GBI enforce constraints**
  without compromising the ***F1 score*** of the system?
- **Q3.** *Can* ***GBI generalize*** over multiple baseline model?
- **Q4.** Performance of GBI on out-of-domain dataset?

Research Questions

- **Q1.** Does GBI actually ***enforce constraints*****?**
- **Q2.** Does **GBI enforce constraints**
  without compromising the ***F1 score*** of the system?
- **Q3.** *Can* ***GBI generalize*** over multiple baseline model?
- **Q4.** How is the performance of ***GBI on out-of-domain*** dataset?

Back to SRL

- Revisiting *Semantic Role Labeling (SRL)*
  - Finding (Agent (arg0), Patient (arg1)) given verb

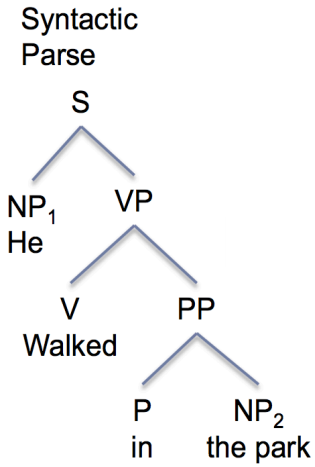


Syntactic constraint on SRL

- Given syntax tree, span candidates were ***limited*** by syntactic information.

X (Input)

"He walked in the park"



Possible span candidate

(He), (walked), (in the park),
(walked in the park),
(He walked in the park)

(He) (walked) (in the park)

~~(He) (walked in)(the park)~~

~~(He walked) (in the park)~~

~~(He walked in the)(park)~~

(He walked in the park)

Wrong spans marked with strike

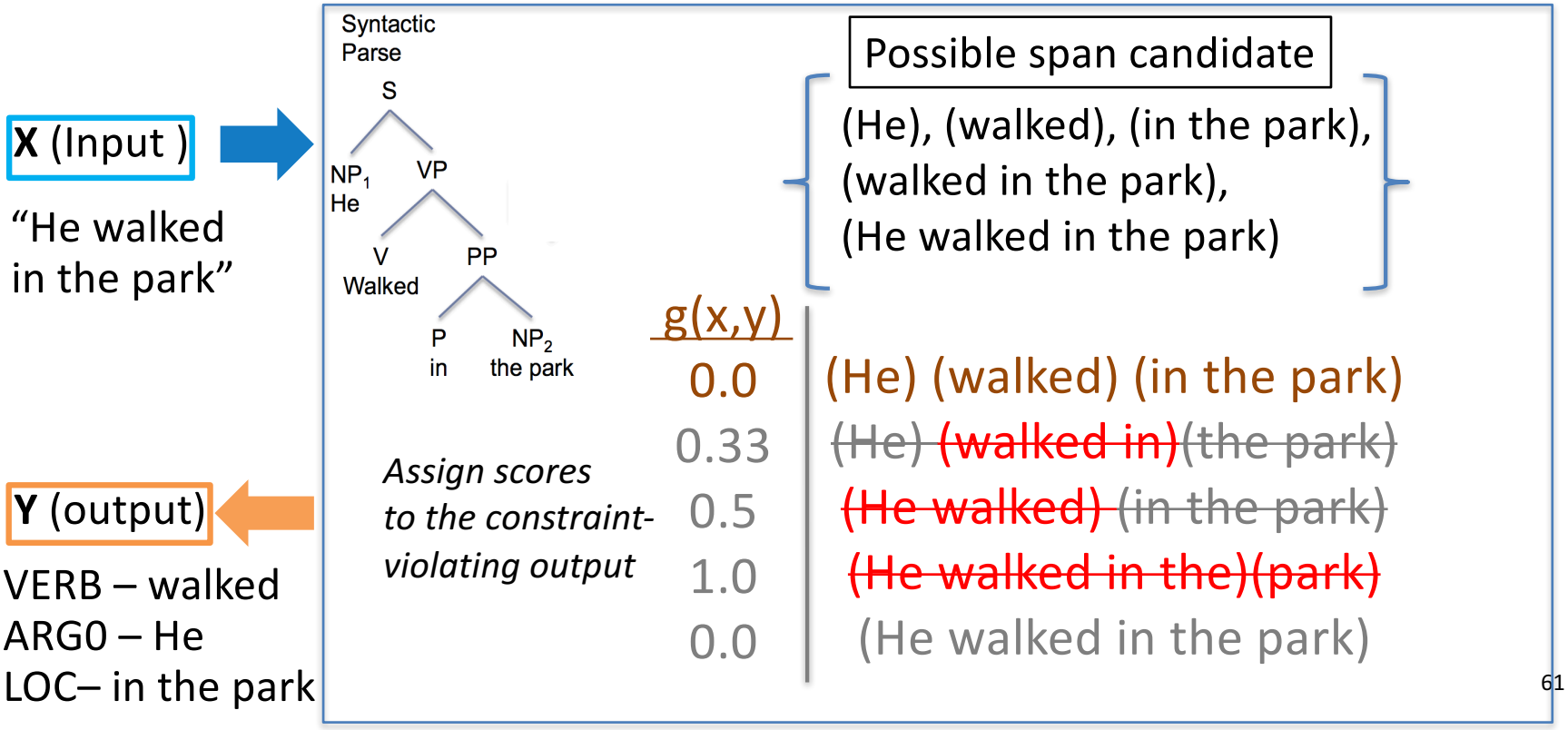
Constraint on SRL

- srl-spans(**y**): spans of predicted SRL output **y**
- parse-spans(**x**): parsing constituents without labels.
- disagreeing-spans(**x**,**y**) = $\{span_i \in \text{srl-spans}(\mathbf{y}) \mid span_i \notin \text{parse-spans}(\mathbf{x})\}$
- constraint function $g(\mathbf{x},\mathbf{y}) \geq 0$ on SRL

$$g(\mathbf{x}, \mathbf{y}) = \frac{|\text{disagreeing-spans}(\mathbf{x}, \mathbf{y})|}{|\text{srl-spans}(\mathbf{x}, \mathbf{y})|}$$

Constraint on SRL

- Constraint (violation) scores on previous exmaple



GBI on SRL: Experiments

- OntoNotes v5.0, CoNLL-2012 shared task
  (278k train, 38.3k dev, 25.6k test)

- Gold predicates and gold parse constituents are given
  (~10% noisy annotations) .

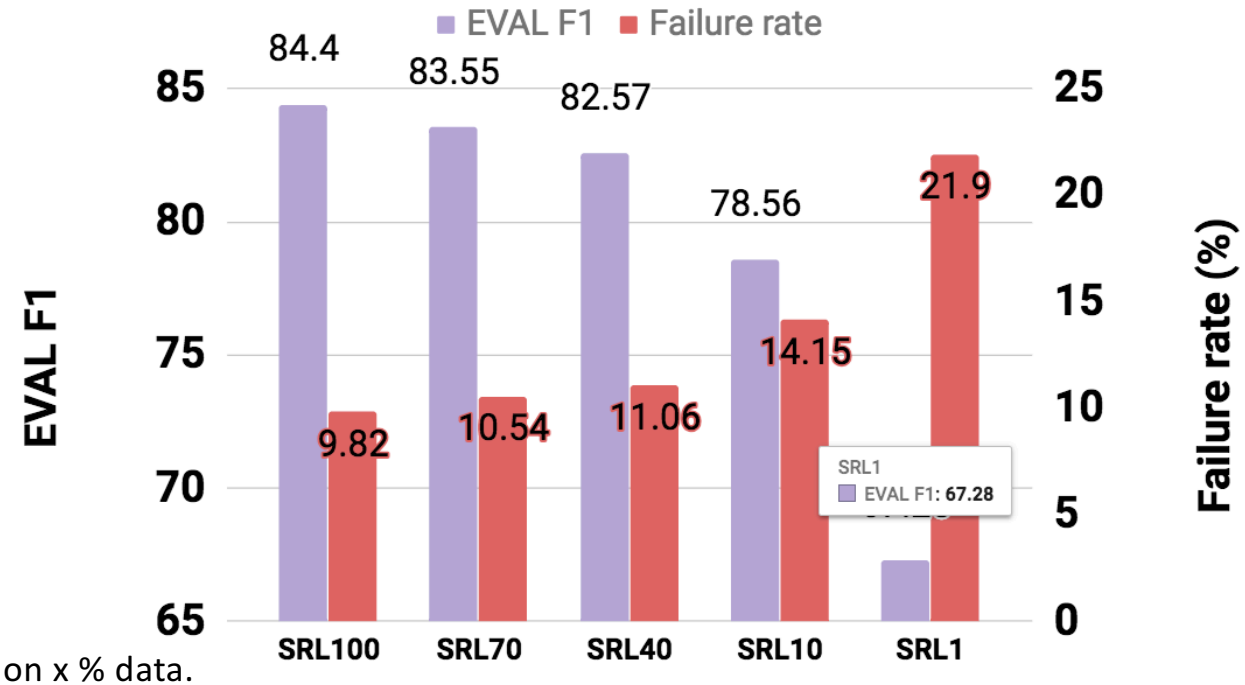
- Baseline architecture
  - multi-layer highway bi-directional LSTM (ELMo embedding)

GBI on SRL: Experiments

- Statistics
  - **Failure set:** Set of instances where SRL output violates simple constraint.
  - **Failure rate (%)**: Relative size of failure set compared to whole test set.
  - **Conv rate (%):** The relative portion of the failure set converted to constraint-satisfying output after GBI.

GBI on SRL: Experiments

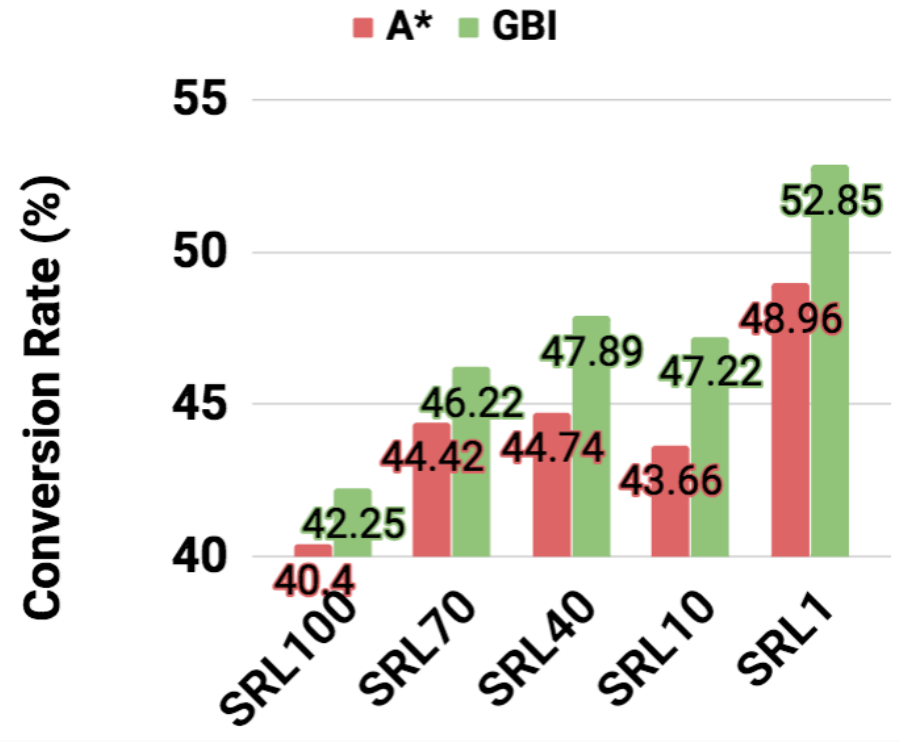
- Trained 5 networks with different level of performance by using different amounts of data (1%,10%, ⋯,100%)



SRL x : SRL system trained on x % data.

GBI on SRL: Experiments

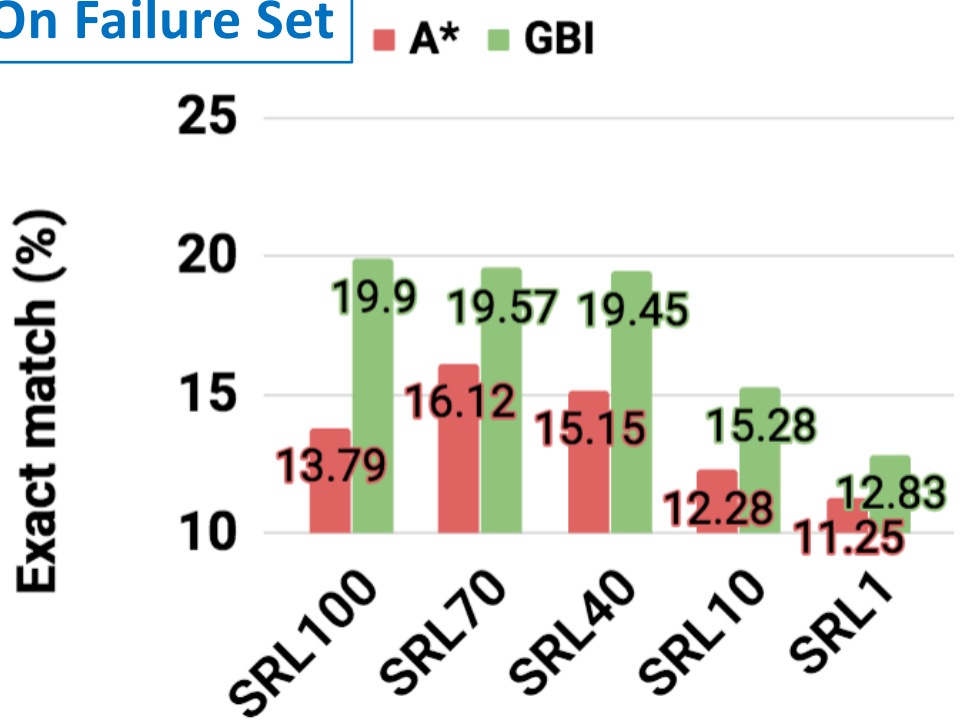
- Q3: GBI enforces constraints. (much better than A*)



GBI on SRL: Experiments

- Q3: GBI enforces constraints **without hurting** the system.

On Failure Set

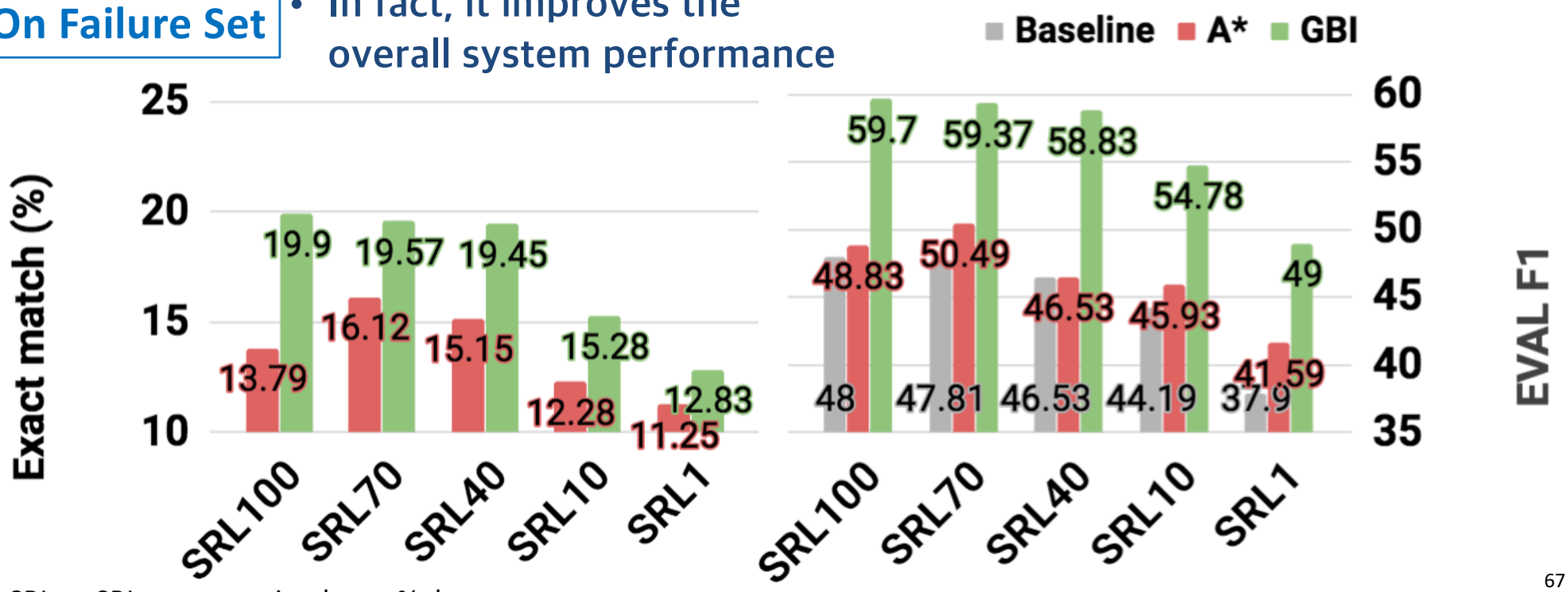


SRL x : SRL system trained on x % data.

GBI on SRL: Experiments

- Q3: GBI enforces constraints **without hurting** the system.
- **In fact, it improves the overall system performance**

On Failure Set

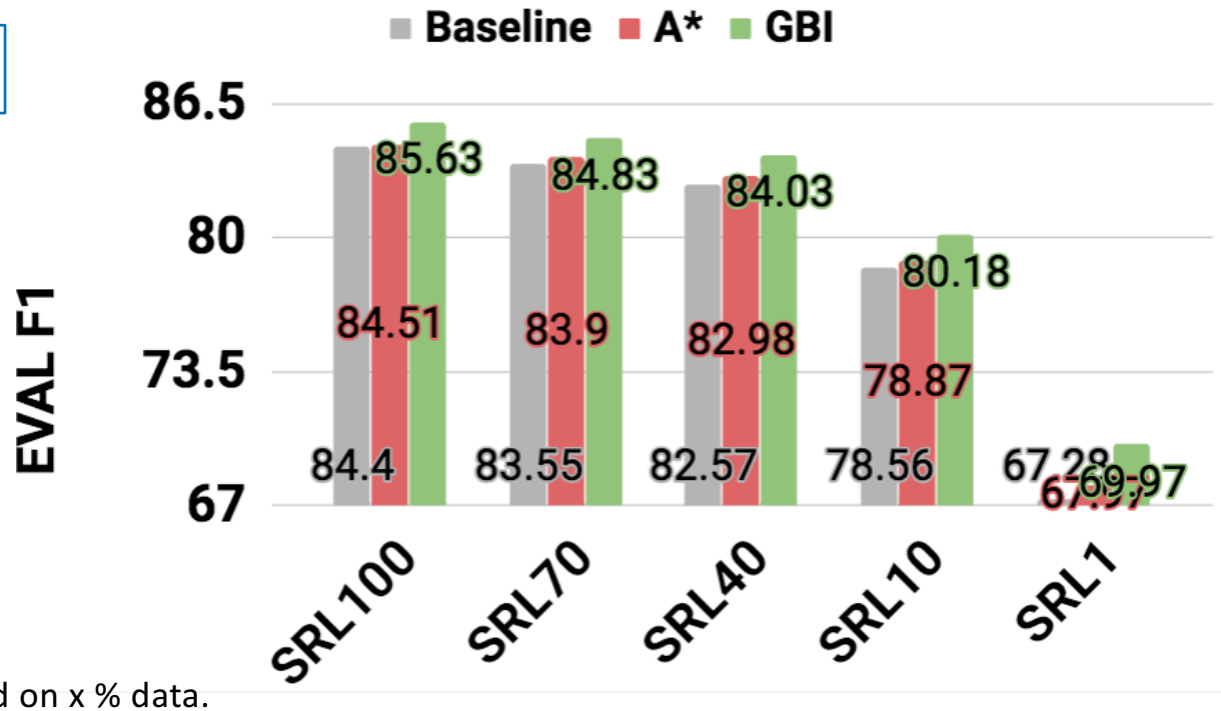


SRL x : SRL system trained on x % data.

GBI on SRL: Experiments

- Q3: GBI enforces constraints **without hurting** the system.
  - **In fact, it improves the overall system performance**

On Total test set

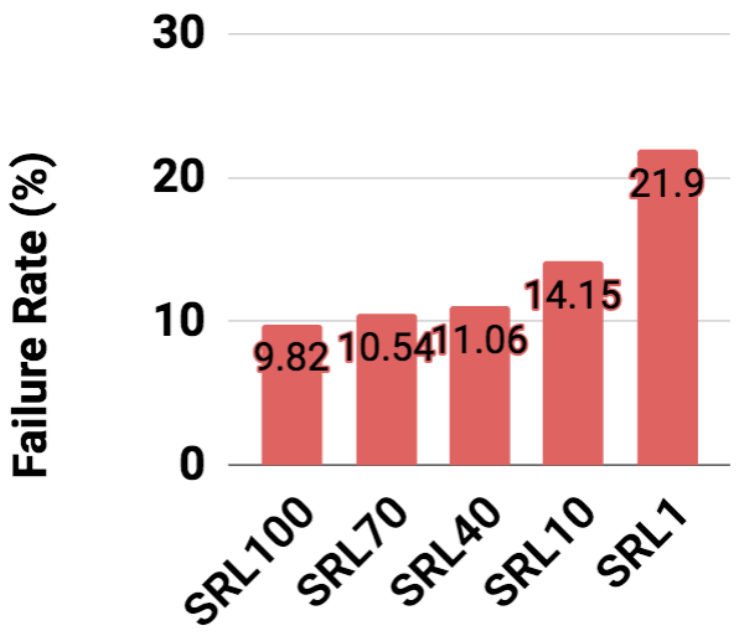


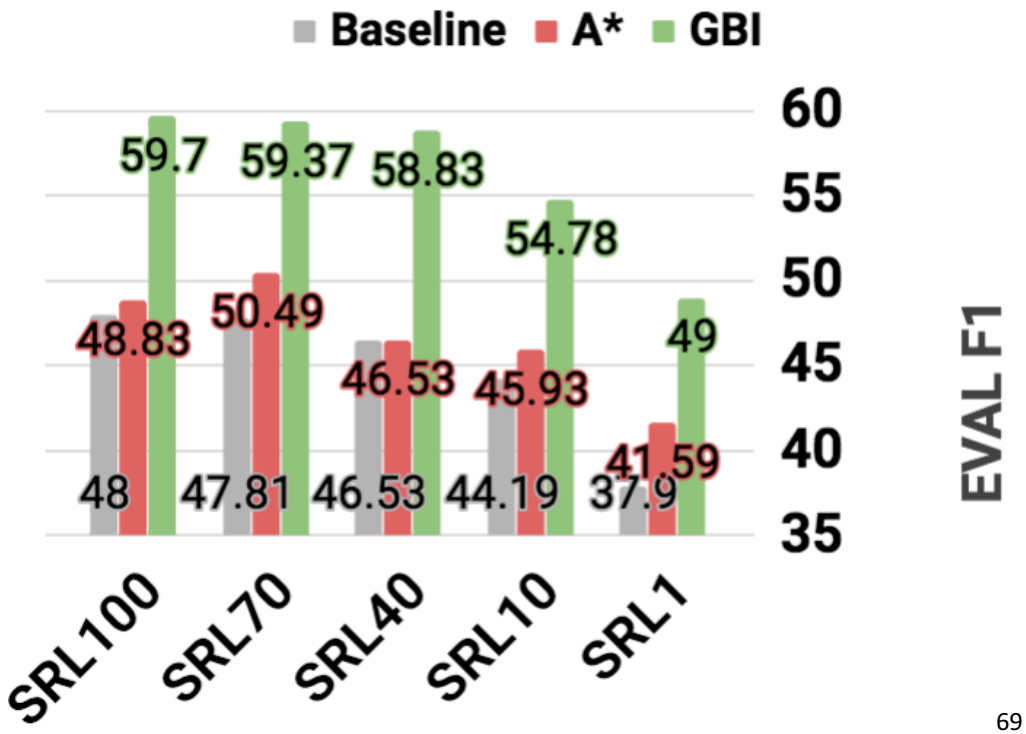
SRL x : SRL system trained on x % data.

GBI on SRL: Experiments

- Q4: Not sensitive to the baseline model performance

On Failure Set





Application 2: Syntactic Parsing

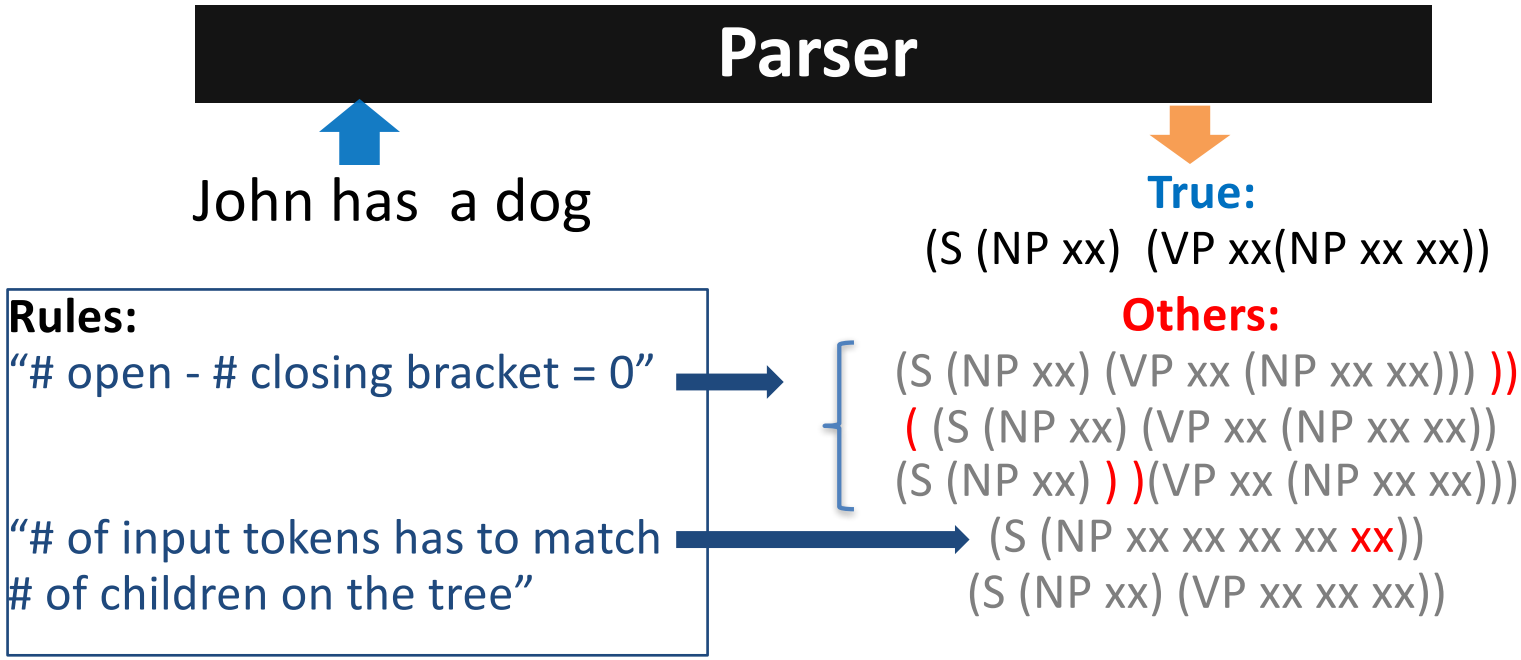
- **By a linguist**
  - Deterministic rule based. (thousands of rules)

- **Statistical methods**
  - $O(N^3)$ and accurate
  - O(N) and inaccurate

- **State of the art:** Neural Networks.
  - Using generic sequence-to-sequence model
  - Using span-based models

Constraints on parsing model

- seq2seq **does not guarantee** "very simple rule".



Constraints function

- **Rules**
  - # of input tokens = # of children on output tree
  - # opening brackets = # of closing brackets

- **Constraint function definition (x:input, y: output)**
  - **Simple count of errors normalized by** sum of input, output **length**.
  - Roughly speaking:

$$g(y, \mathcal{L}^x) = \frac{1}{|x| + |y|} \{abs(|x| - |y_{children}|) + |\text{count}_{open} - \text{count}_{close}|\}$$

GBI on parsing: Experiments

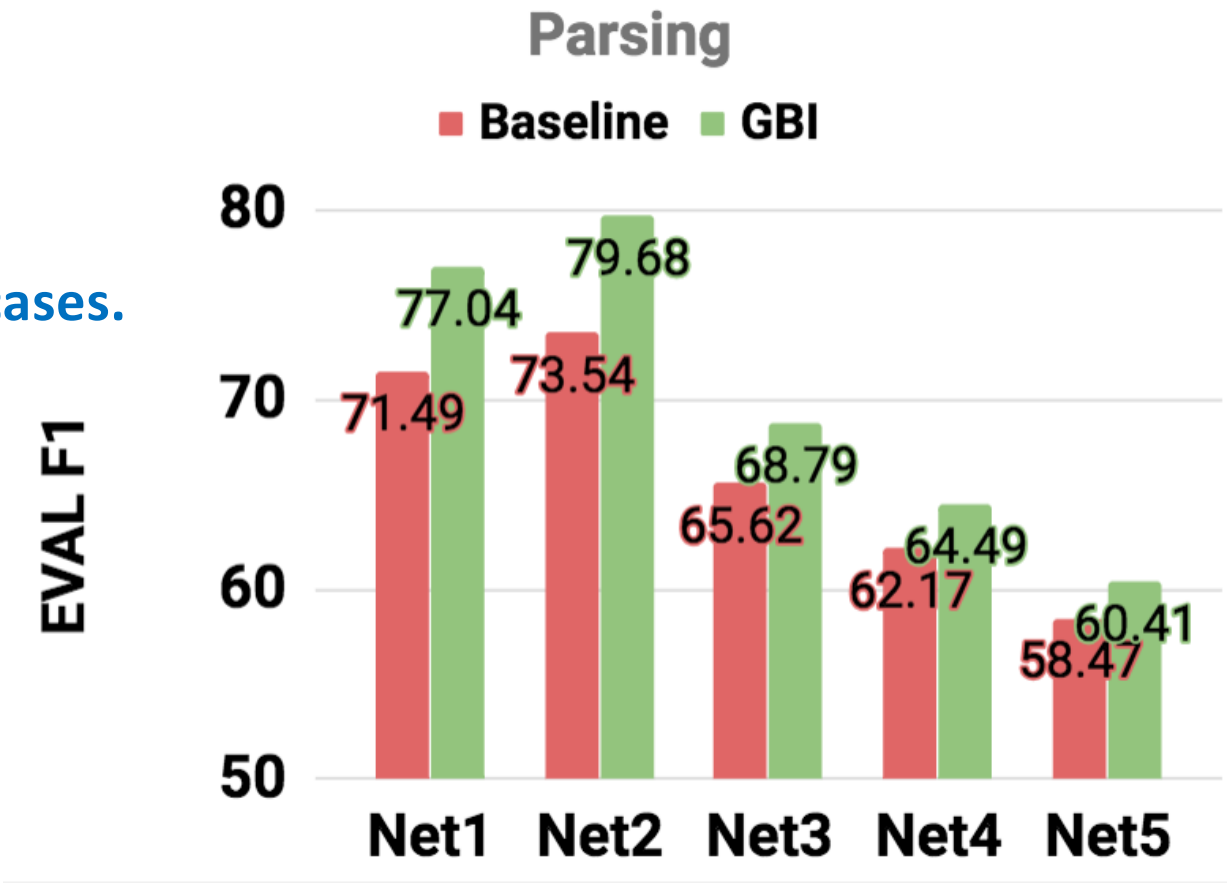
- Trained 5 parsers to answer Q1-4 (especially Q3)
  - Net1-2: GNMT (bi-directional encoder, uni-directional decoder)
  - Net3-5: uni-directional encoder, decoder.
  - Different dropout, used different portion of the dataset.

name	F1		hyper-parameters			data
	BS-9	greedy	hidden	layers	dropout	(%)
Net1	87.58	87.31	128	3	0.5	100%
Net2	86.63	86.54	128	3	0.2	100%
Net3	81.26	78.32	172	3	no	100%
Net4	78.14	74.53	128	3	no	75%
Net5	71.54	67.80	128	3	no	25%

GNMT: Wu et al, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, 2016

GBI on parsing: Experiments

Similar trend.
Again, GBI
Improves all cases.



GBI on parsing: Experiments

- Q4: Not sensitive to the baseline model performance.
- Not only that, **GBI is also compatible with various decoding methods.**

Net	Infer method	Failure (n/2415)	Conv rate	F1 (Failure set) before	F1 (Failure set) after
Net3	Greedy	317	79.81	65.62	68.79 (+3.14)
	Beam 2	206	87.38	66.61	71.15 (+4.54)
	Beam 5	160	87.50	67.5	71.38 (+3.88)
	Beam 9	153	91.50	68.66	71.69 (+3.03)
Net4	Greedy	611	88.05	62.17	64.49 (+2.32)
	Beam 2	419	94.27	65.40	66.65 (+1.25)
	Beam 5	368	92.66	67.18	69.4 (+2.22)
	Beam 9	360	93.89	67.83	70.64 (+2.81)
Net5	Greedy	886	69.86	58.47	60.41 (+1.94)
	Beam 2	602	82.89	60.45	61.35 (+0.90)
	Beam 5	546	81.50	61.43	63.25 (+1.82)
	Beam 9	552	80.62	61.64	62.98 (+1.34)

GBI on parsing: Experiments

- Q4: Not sensitive to the baseline model performance.
- Not only that, **GBI is also compatible with various decoding methods.**

More details,
Please refer to
the thesis proposal.

Net	Infer method	Failure (n/2415)	Conv rate	F1 (Failure set) before	F1 (Failure set) after
Net3	Greedy	317	79.81	65.62	68.79 (+3.14)
	Beam 2	206	87.38	66.61	71.15 (+4.54)
	Beam 5	160	87.50	67.5	71.38 (+3.88)
	Beam 9	153	91.50	68.66	71.69 (+3.03)
Net4	Greedy	611	88.05	62.17	64.49 (+2.32)
	Beam 2	419	94.27	65.40	66.65 (+1.25)
	Beam 5	368	92.66	67.18	69.4 (+2.22)
	Beam 9	360	93.89	67.83	70.64 (+2.81)
Net5	Greedy	886	69.86	58.47	60.41 (+1.94)
	Beam 2	602	82.89	60.45	61.35 (+0.90)
	Beam 5	546	81.50	61.43	63.25 (+1.82)
	Beam 9	552	80.62	61.64	62.98 (+1.34)

GBI on transducer

- *Transduction (toy)*
  - A transducer (T) is a function from a 'source language' (L_s) to a 'target language' (L_T).

Experiment Result
- 65.2% conversion rate
- On Failure set
  - Before-GBI: 75.2%
  - After-GBI: 82.4%

$$T : \mathcal{L}_S \rightarrow \mathcal{L}_T$$

T: az → aaa
bz → zb

S: azazbzazbzbazbzbzbzbzbz

T: aaaaaazbaaazbzbaaazbzbzbzbzb

constraint:
3 x (# a in **S**) = # a in **T**

GBI on transducer

- *Transduction (toy)*
  - A transducer (T) is a function from a 'source language' (L_s) to a 'target language' (L_T).

Experiment Result
- 65.2% conversion rate
- On Failure set (accuracy)
  - Before-GBI: 75.2%
  - After-GBI: 82.4%

More details,
Please refer to
the thesis proposal.

$$T : \mathcal{L}_S \rightarrow \mathcal{L}_T$$

T: az → aaa
bz → zb

S: azazbzazbzbazbzbzbzbzbz

T: aaaaaazbaaazbzbaaazbzbzbzbzb

constraint:
3 x (# a in **S**) = # a in **T**

GBI: extra considerations

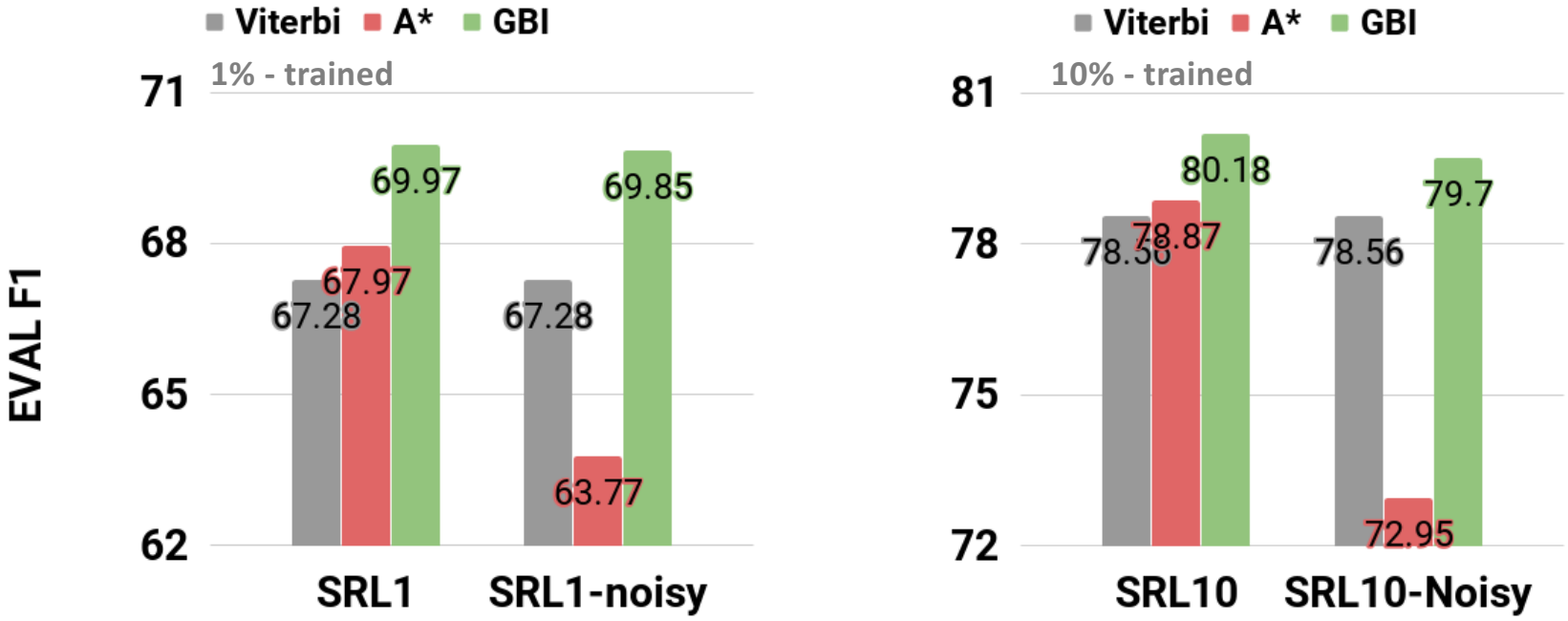
- Robustness to noisy constraint.
- Q5) out-of-domain performance.

Robustness of GBI

- SRL has about 10% of the span mismatch between annotated SRL output (propbank) & parse tree (treebank)

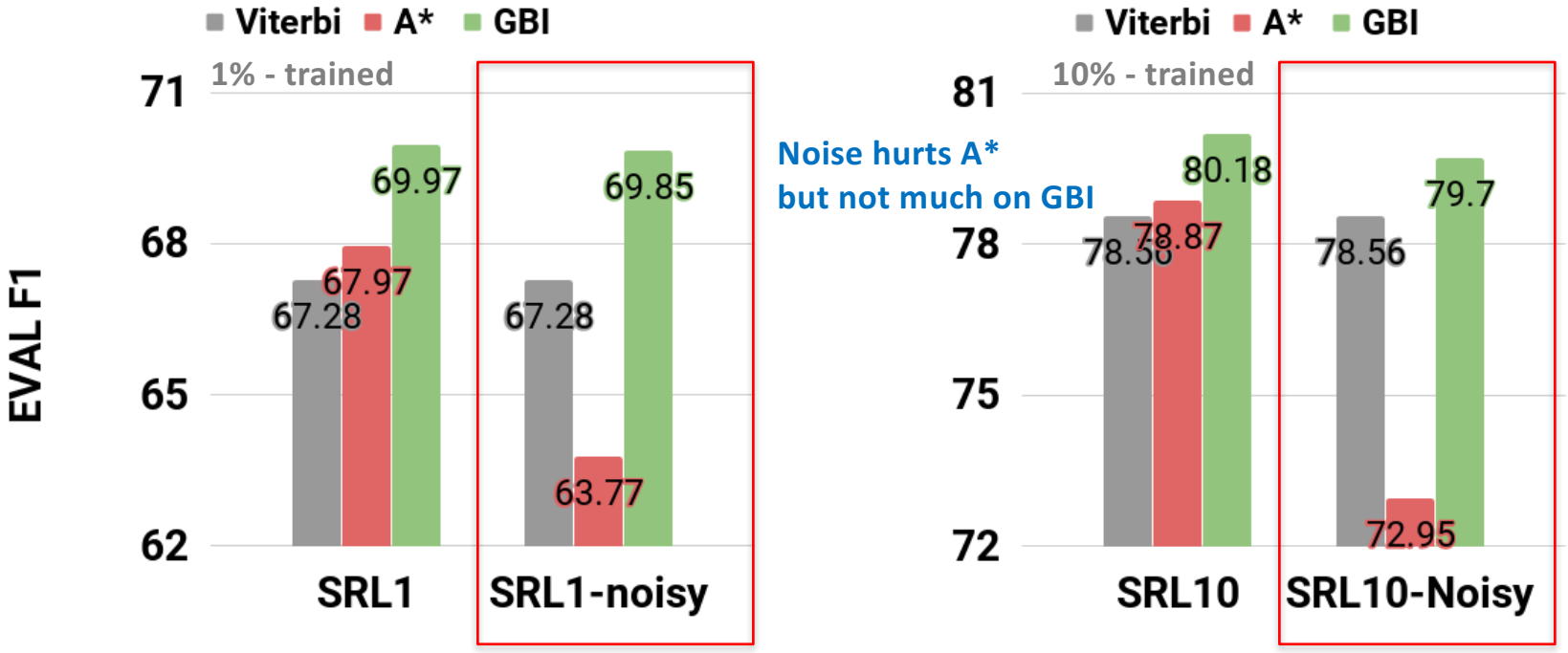
Robustness of GBI

- SRL has about 10% of the span mismatch between annotated SRL output (propbank) & parse tree (treebank)



Robustness of GBI

- SRL has about 10% of the span mismatch between annotated SRL output (propbank) & parse tree (treebank)

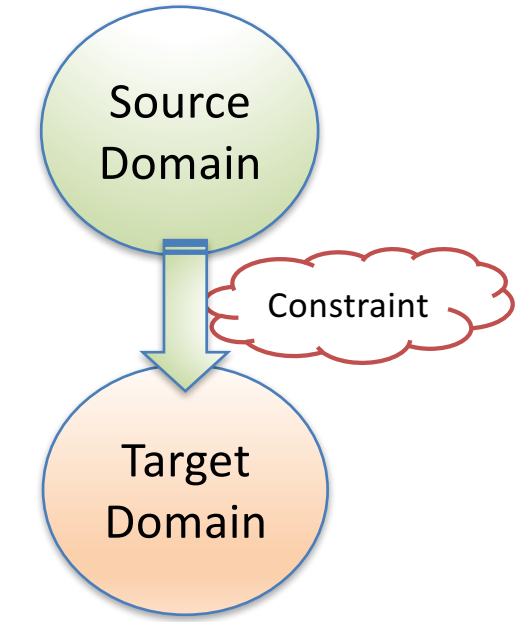


GBI on out-of-domain data (Q5)

- CoNLL2012
  - NewsWire (NW)
  - Broadcast Conversation (BC)
  - Broadcast News (BN)
  - Pivot Corpus (PT)
  - Telephone Conversation (TC)
  - Weblogs (WB)

GBI on out-of-domain data

- CoNLL2012
  - NewsWire (NW) which includes WSJ
  - Broadcast Conversation (BC)
  - Broadcast News (BN)
  - Pivot Corpus (PT)
  - Telephone Conversation (TC)
  - Weblogs (WB)



Can constraint injection transfer to other domains?

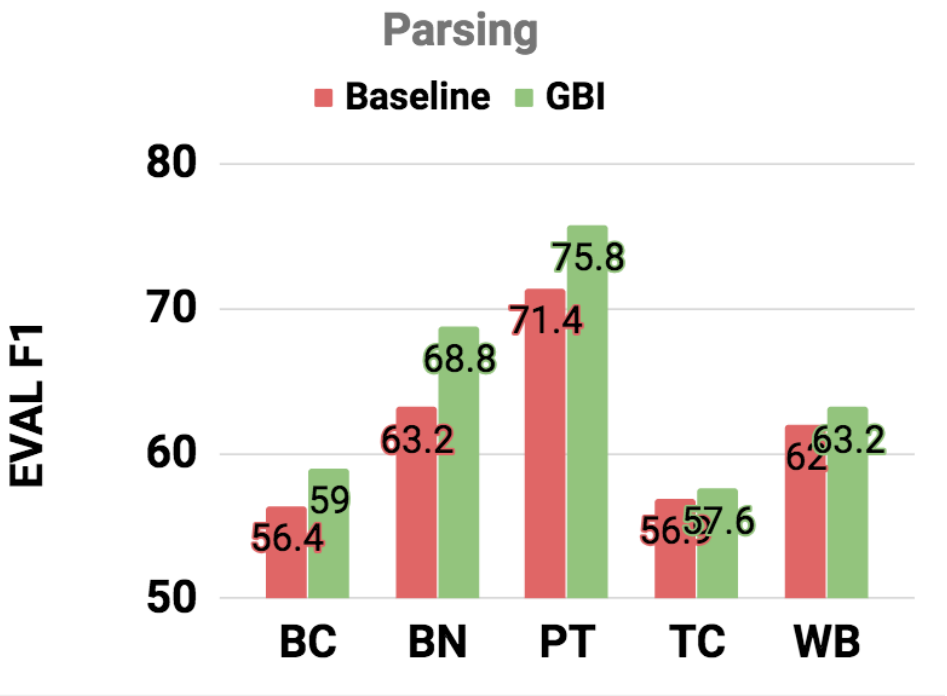
GBI on out-of-domain data

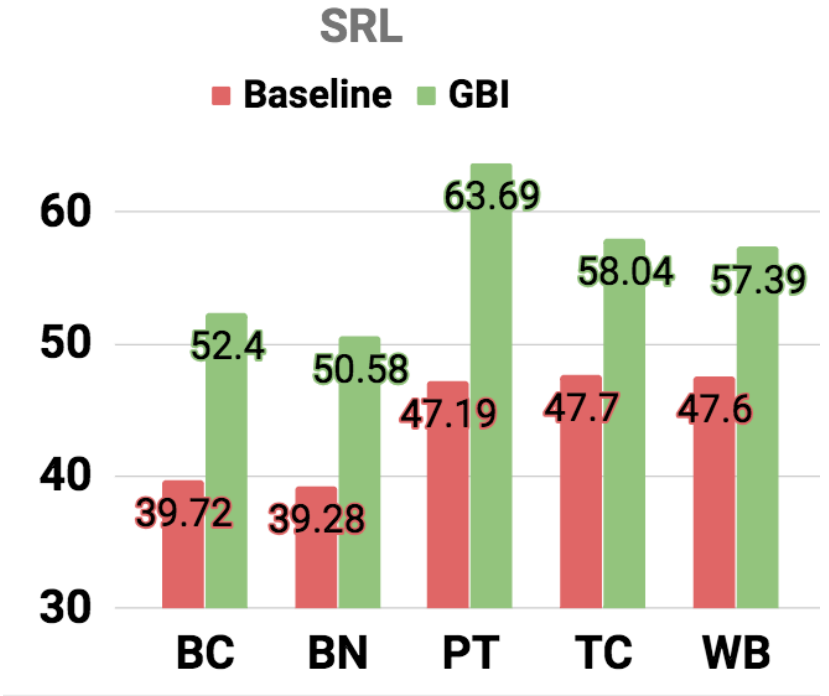
- Generally higher failure rate on out-of-domain evaluation compared to in-domain (parsing: 11.9 %, SRL: 18.1%).

Genre \Task	Failure rate (%)	Conversion rate (%)	F1 on failure set	
			before	after
		Syntactic Parsing		
Broadcast Conversation (BC)	19.3	98.8	56.4	59.0 (+2.6)
Broadcast News (BN)	11.7	98.1	63.2	68.8 (+5.6)
Pivot Corpus (PT)	9.8	97.8	71.4	75.8 (+4.4)
Telephone Conversation (TC)	10.1	86.2	56.9	57.6. (+0.7)
Weblogs (WB)	17.6	95.3	62.0	63.2 (+1.2)
Genre \Task		***SRL***		
Broadcast Conversation (BC)	26.86	53.88	39.72	52.4 (+12.68)
Broadcast News (BN)	18.51	55.19	39.28	50.58 (+11.3)
Pivot Corpus (PT)	10.01	62.34	47.19	63.69 (+16.5)
Telephone Conversation (TC)	19.09	54.62	47.7	58.04 (+10.34)
Weblogs (WB)	20.32	44.13	47.6	57.39 (+9.39)

GBI on out-of-domain data

- In summary, improves in all cases (reporting on failure set)
  - While parsing is more sensitive to domains.





Roadmap

- Focus of our work
- Applications & constraints
- Gradient-Based Inference formulation
- Research questions & Experiments
- **Conclusion**

Research Questions

- **Q1.** Does **GBI** actually ***enforce constraints***?
  - Yes!
- **Q2.** Does **GBI** ***enforce constraints without compromising*** the quality of the system?
  - Yes. It rather Improves!
- **Q3.** Is **GBI** ***sensitive*** to the ***baseline model***?
  - Not on 3 applications (5 experiment settings)
- **Q4.** ***Performance*** of GBI **on** ***out-of-domain*** dataset?
  - Good!

Thank You!

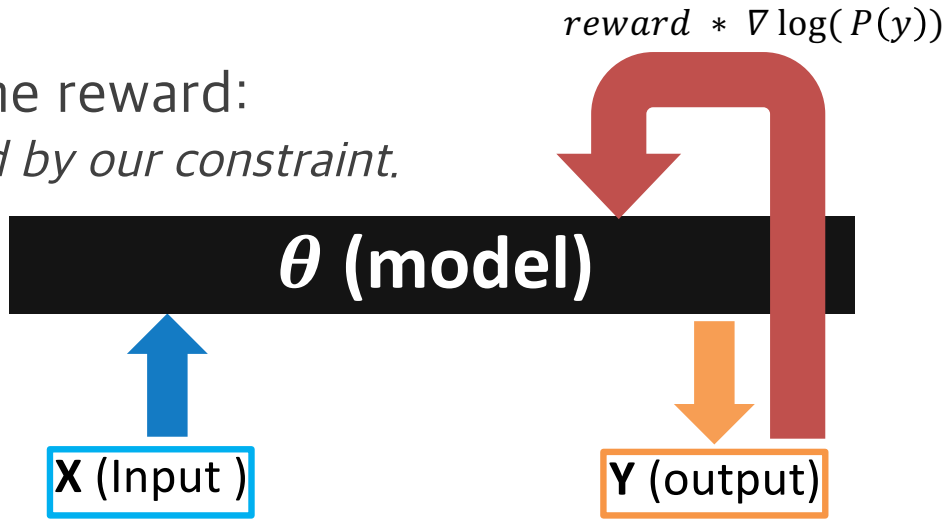
Questions?

[backup] Learning with constraints

- Optimization with constraints

$$\min_{W} \quad f(\mathbf{x}, \mathbf{y}; W)$$
$$s.t. \quad g(\mathbf{y}, \mathcal{L}) = 0$$

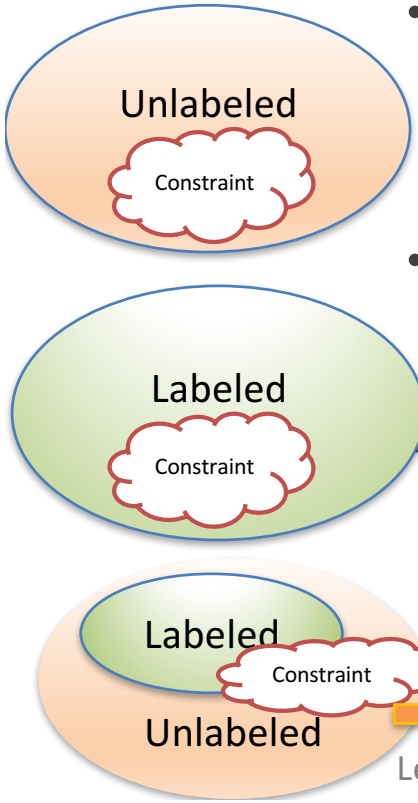
- Learning with the reward:
  - *Reward defined by our constraint.*



Lee & Mehta et al, Towards semi-supervised learning for deep semantic role labeling, EMNLP2018

[backup] Learning with constraints

- Learning with the reward s(x,y)
  - constraint loss (c-loss)



$$s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}; \mathbf{w}^{(t)}) + \|W - W_{\text{pre-train}}\|_2^2$$

- Joint loss

Semi-supervised Joint loss

SRL-***labeled*** data $(\mathbf{x}_m = \mathbf{x}_k)$

$$-\alpha_1 \sum_{i=1}^{|\mathbf{y}|} \log p\,(y_i \mid \mathbf{x}_k; \mathbf{w}) + \alpha_2 s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} \mid \mathbf{x}_m; \mathbf{w}^{(t)})$$

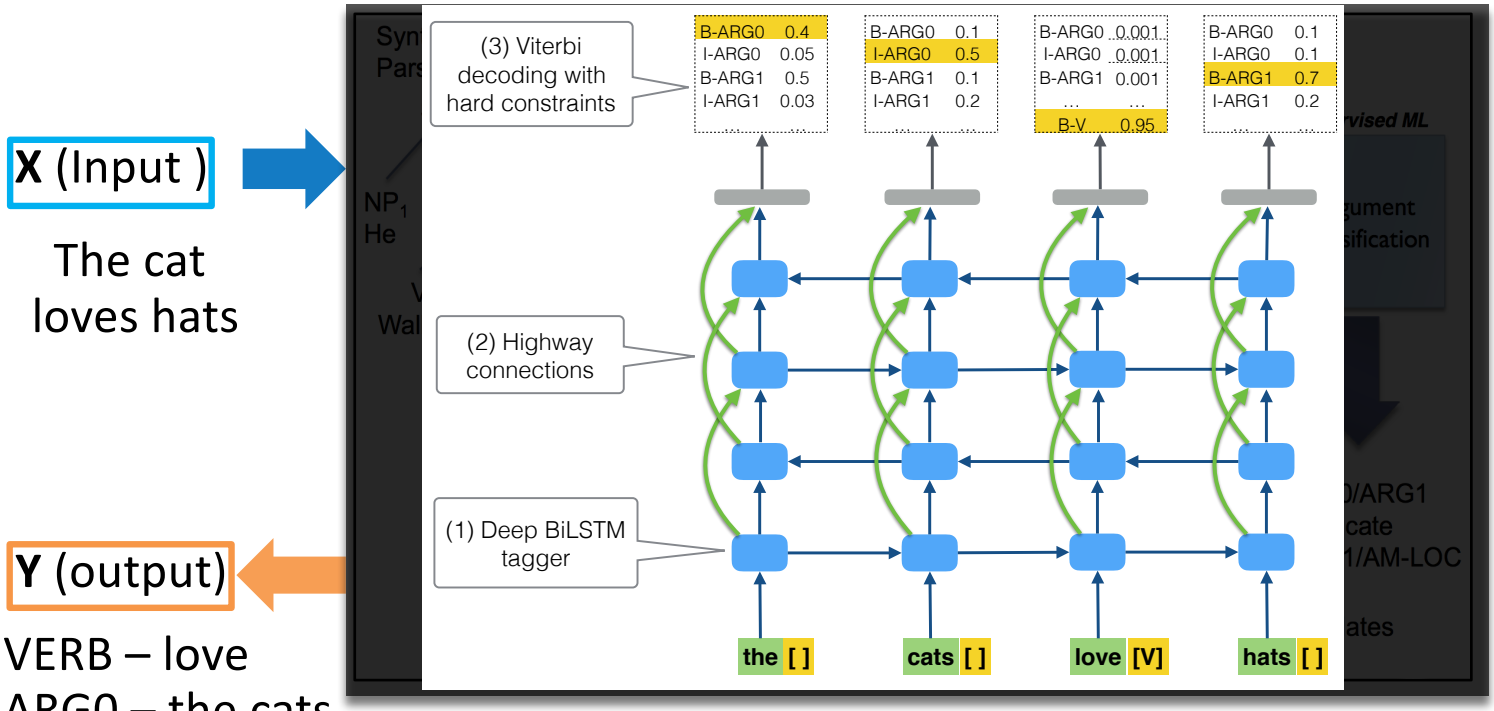
SRL-***labeled*** data

SRL-***unlabeled*** data $(\mathbf{x}_m \neq \mathbf{x}_k)$

Lee & Mehta et al, Towards semi-supervised learning for deep semantic role labeling, EMNLP2018

[Revisit] Complex models

- Semantic Role Labeling

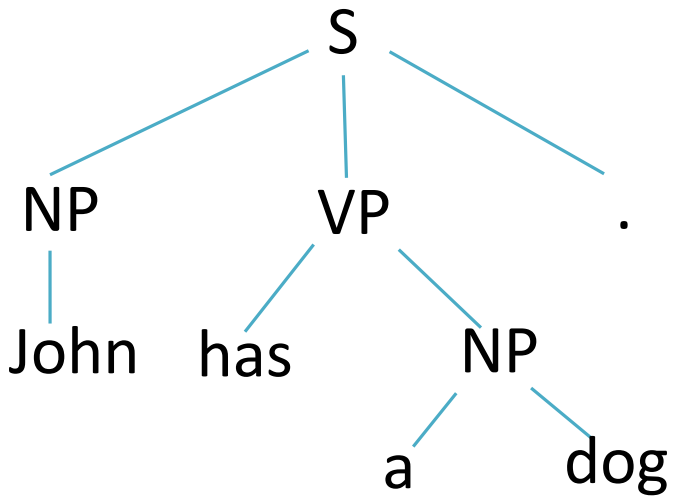


Slide credits: Luheng He

How to express the Grammar Tree as a sequence?

Proposed model

- "As a command" to produce the tree.



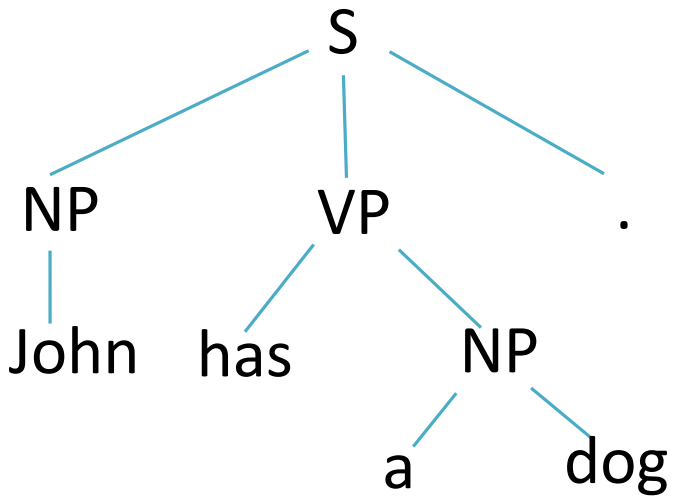
- Shift-Reduce syntax
  - Bottom-up approach
  - Similar to how human minds work.

How to express the Grammar Tree as a sequence?

Proposed model

- "As a command" to produce the tree.

s R-1-NP s s s R-2-NP R-2-VP s R-3-S

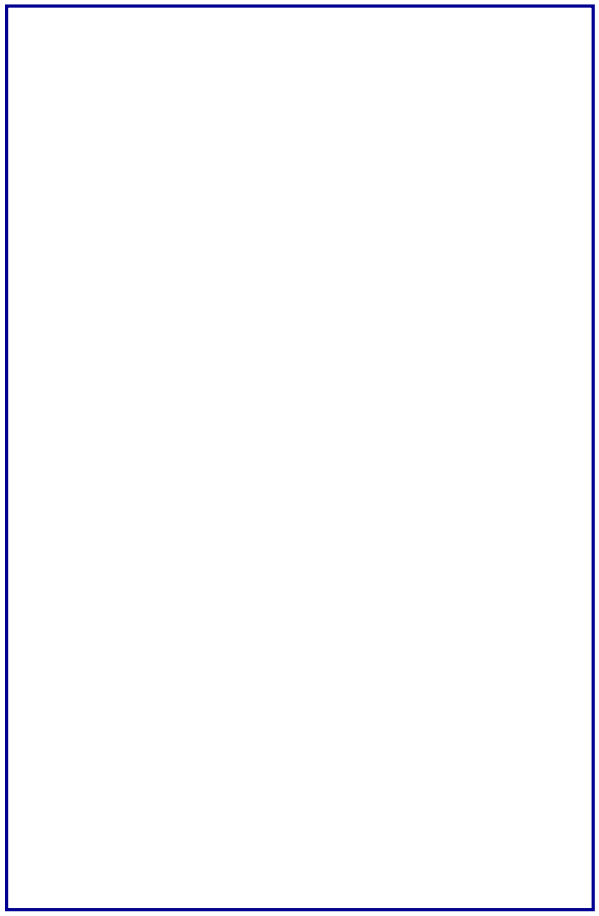


- Shift-Reduce syntax
  - **s** corresponds to "push" in stack.
  - **R-#-XX** corresponds to "pop"-ing that **#** of elements , merging them together to **XX** and pushing it back

Shift-Reduce syntax

Proposed model

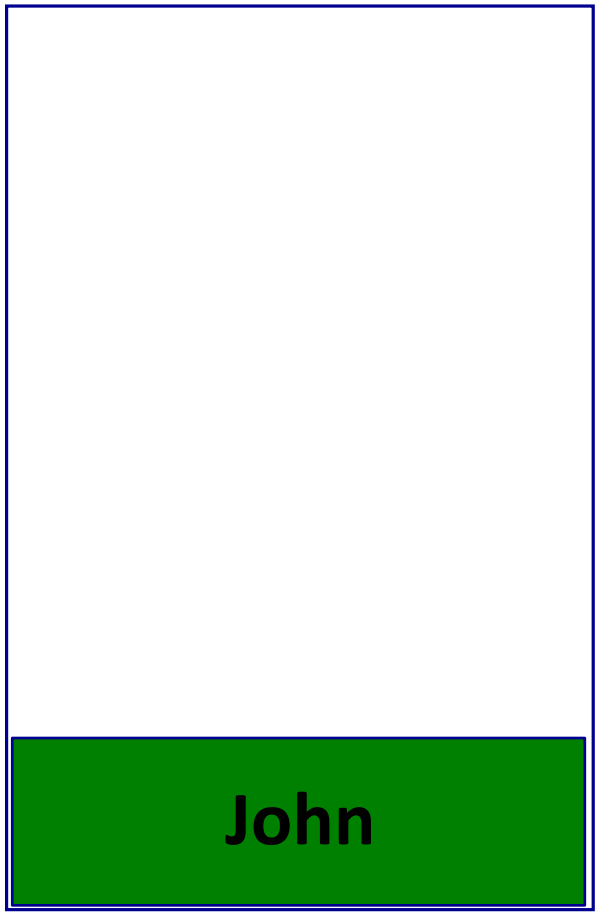
- Original sentence
  John has a dog .
- Shift-Reduce command
  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

Proposed model

- Original sentence
  John has a dog .
- Shift-Reduce command
  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

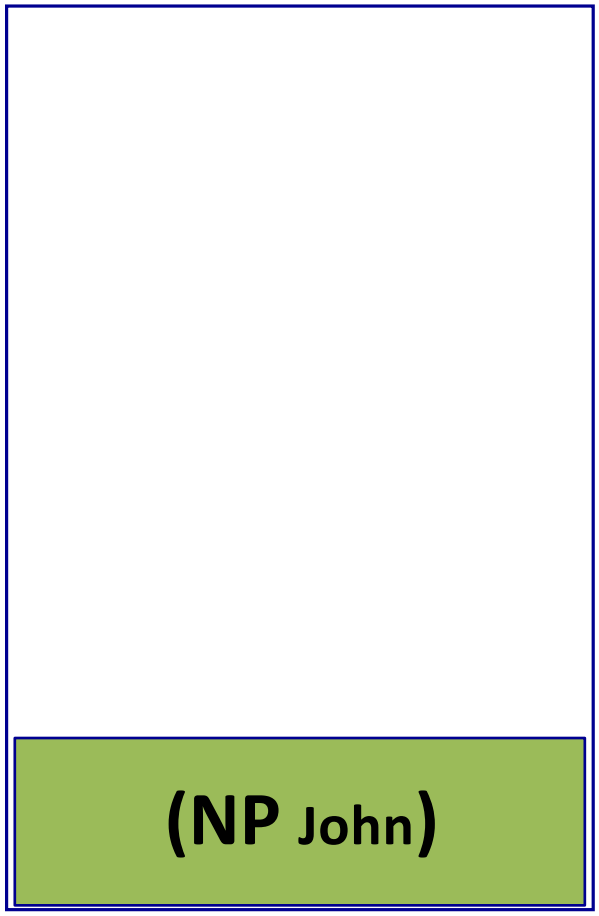
Proposed model

- Original sentence

  John has a dog .

- Shift-Reduce command

  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

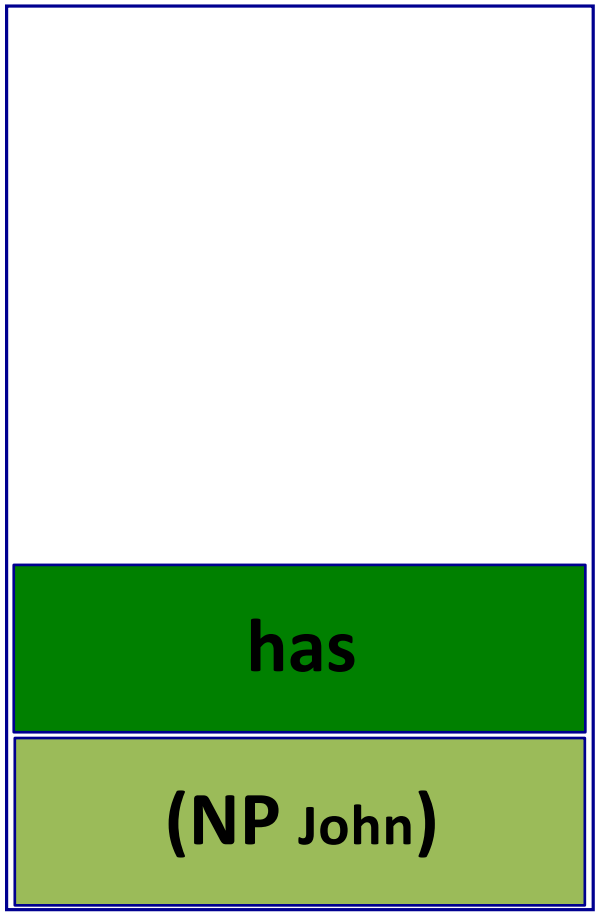
Proposed model

- Original sentence

  John has a dog .

- Shift-Reduce command

  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

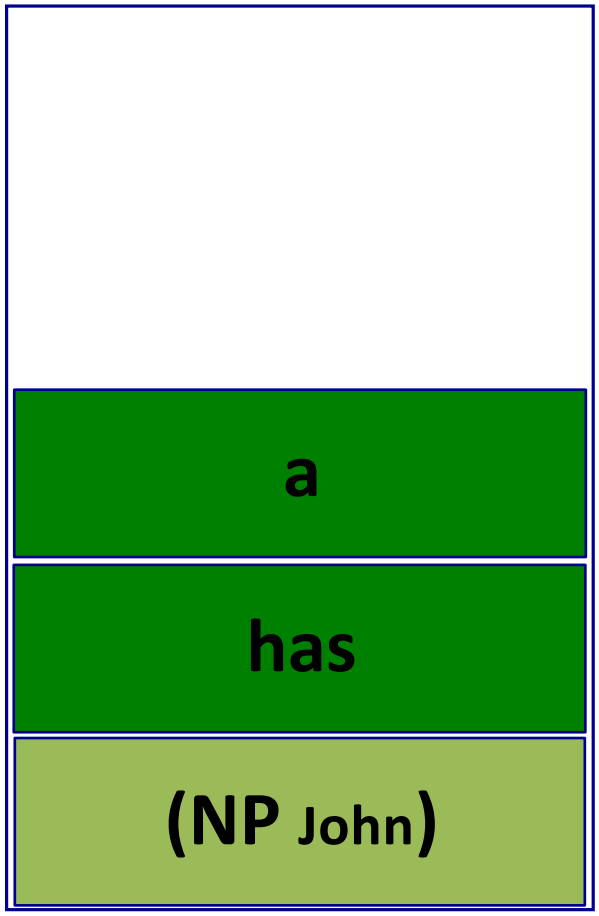
Proposed model

- Original sentence

  John has a dog .

- Shift-Reduce command

  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

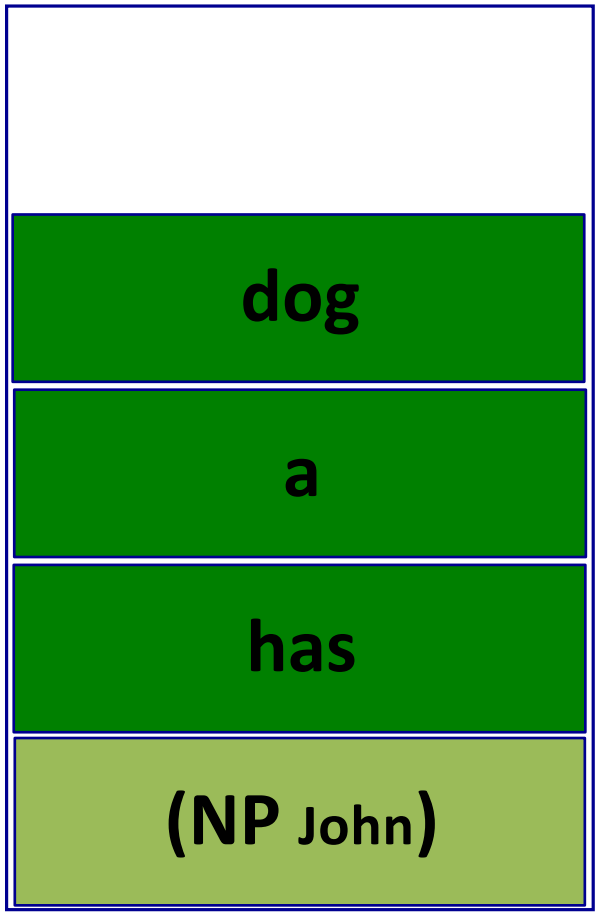
Proposed model

- Original sentence

  John has a dog .

- Shift-Reduce command

  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

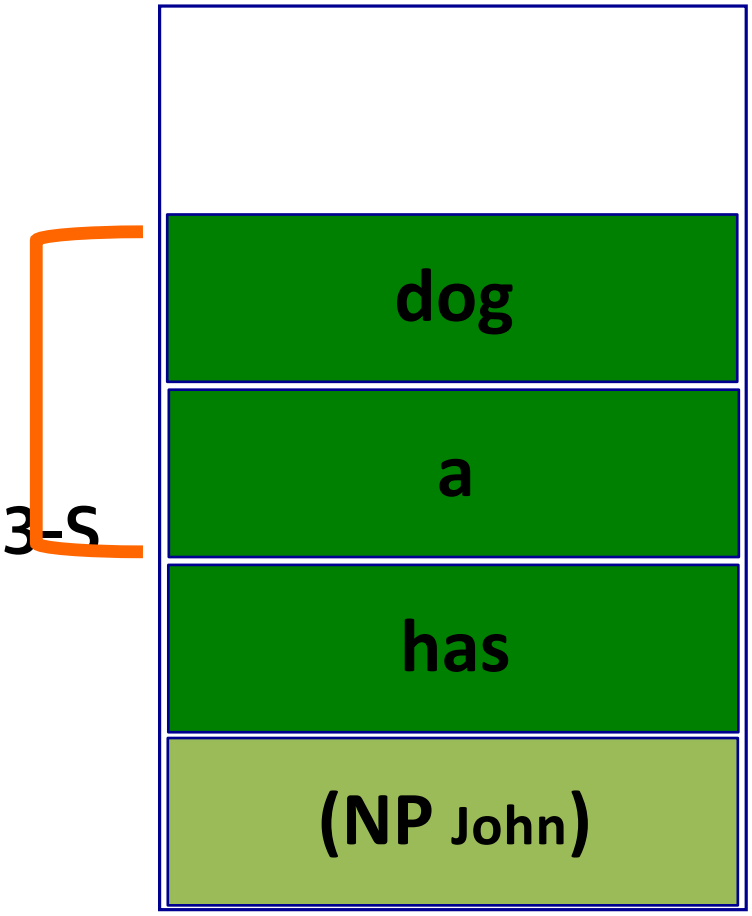
Proposed model

- Original sentence

  John has a dog .

- Shift-Reduce command

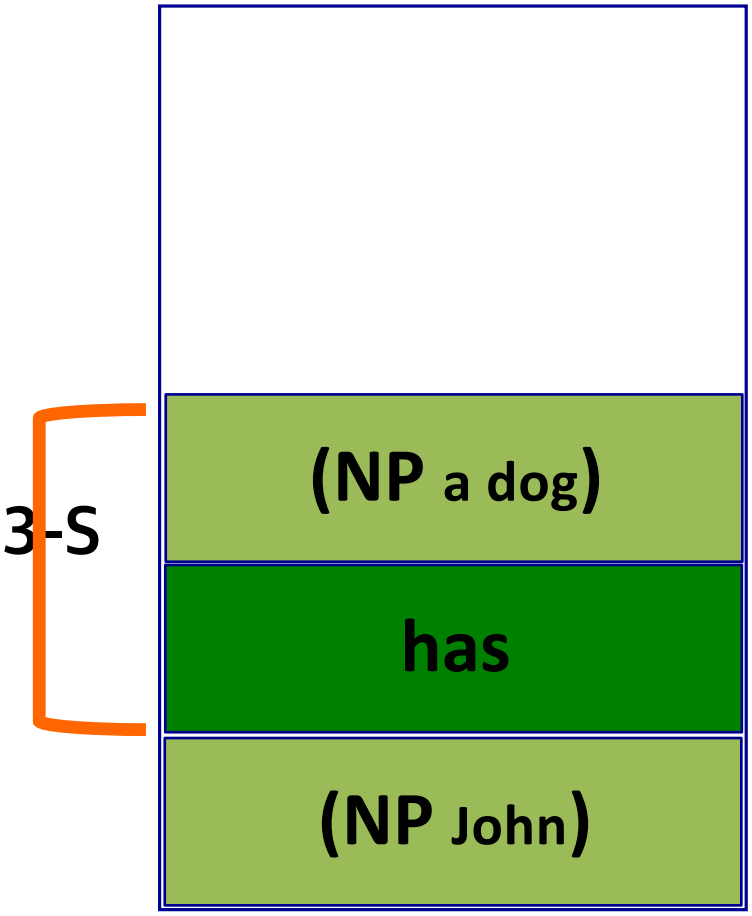
  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

Proposed model

- Original sentence
  John has a dog .
- Shift-Reduce command
  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

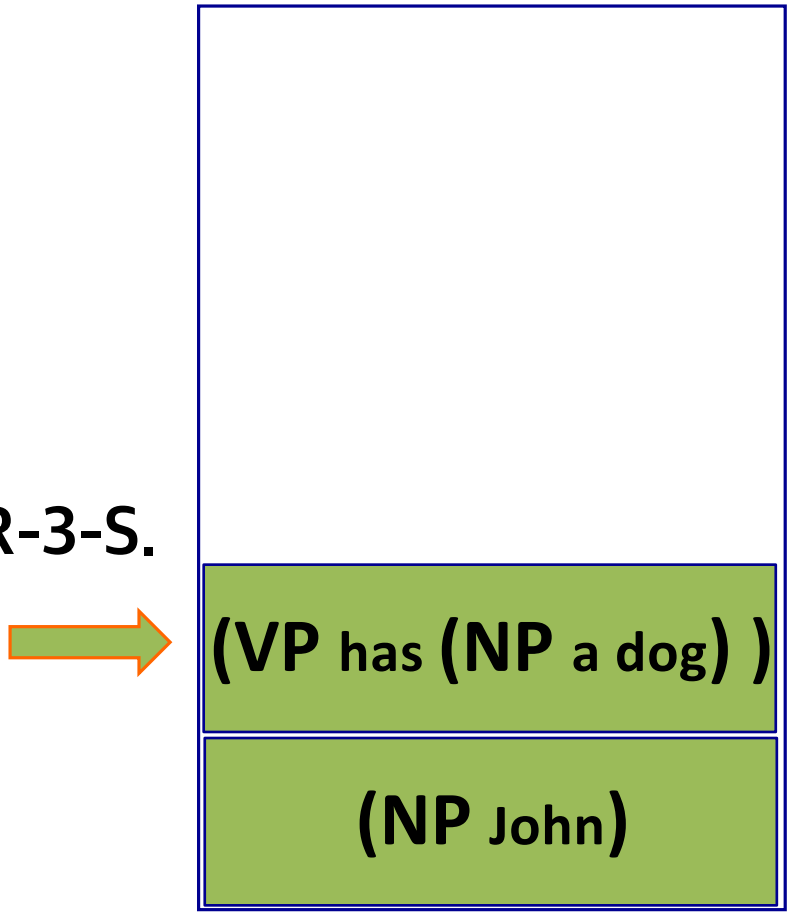
Proposed model

- Original sentence

  John has a dog .

- Shift-Reduce command

  s R-1-NP s s s R-2-NP R-2-VP s R-3-S.



Shift-Reduce syntax

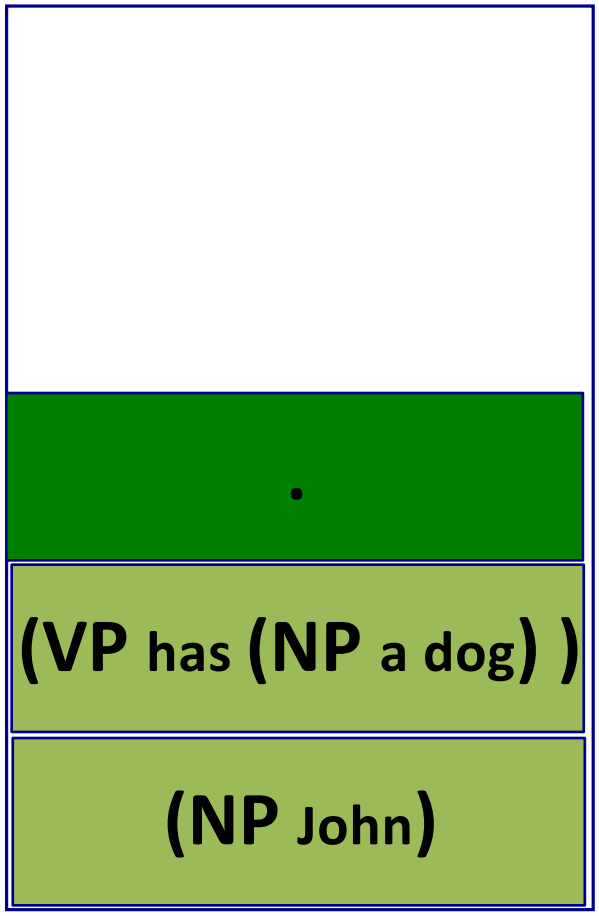
Proposed model

- Original sentence

  John has a dog .

- Shift-Reduce command

  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

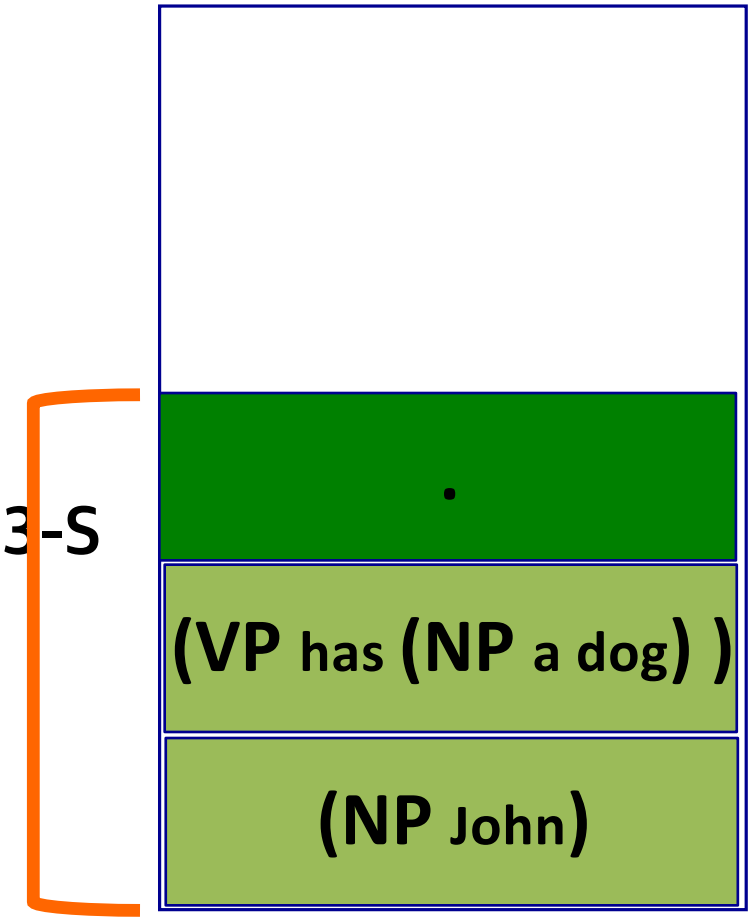
Proposed model

- Original sentence

  John has a dog .

- Shift-Reduce command

  s R-1-NP s s s R-2-NP R-2-VP s R-3-S



Shift-Reduce syntax

Proposed model

- Original sentence

  John has a dog

- Shift-Reduce command

  s R-1-NP s s s R-2-NP R-2-VP s R-3-S

(S (NP John) (VP has (NP a dog)) .)

GBI on

GBI case study

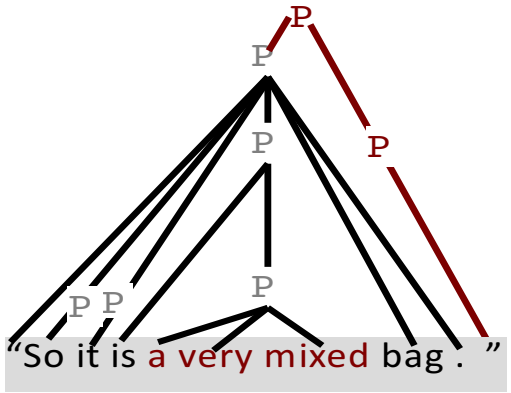
Case study of GBI on parsing

- Comparison to decoder with constraints on each step.
- **Unconstrained decoding + Error signal propagation** can be more powerful.

Constraints in decoder

ssr!sr!ssssrrr!rr!ssrrrrrr!sr!rr!

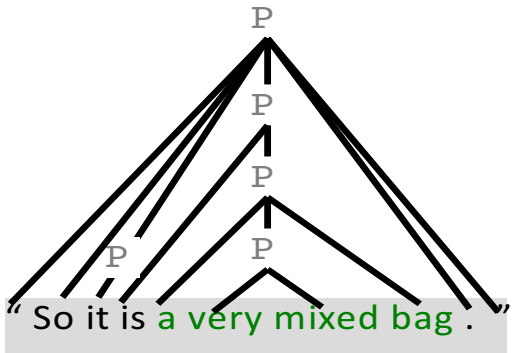
(“(so) (it) (is (a very mixed)) bag .) (”))



Our method
(recovered true)

sssr!ssssr!srrr!rr!ssrrrrrr!

(“ so (it) (is (a (very mixed) bag)) . ”)



GBI extra experiment [WIP]

- SRL has the following constraints
  - BIO - captured by vitrebi
  - Syntactic consistency
  - U,C,R violation
    - Unique core (U) / Continuation (C) / Reference (R) roles

- SOTA models have UCR violations
- GBI successfully reduces U,C,R violations.
  - No F1 score improvement, but the side effect of U,C,R is still unstudied as SRL output is used as an input to the following system.

- WORK IN PROGRESS

GBI on Transduction [WIP]

- Application 3: Transduction problem
  - Definition
    - A transducer T : L_s → L_t is a function from a source language to a target language
  - Transduction itself is a difficult problem.
  - We use it as an example of constraint with seq2seq problem, when you know the aggregate information.
  - Transduction rule: az → aaa

GBI on Transduction

- Case study of GBI on transduction problem.

azazbzazbzbzazbzbzbzbzbz ⟶ aaaaaazbaaazbzbaaazbzbzbzbzb

iteration	output	loss	accuracy
0	aaaaaazbaaazbaaazbzbzbzbaaazb	0.2472	66.7
1	aaaaaazbaaazbaaazbzbzbzbaaazb	0.2467	66.7
2	aaaaaazbaaazbaaazbzbzbzbaaazb	0.2462	66.7
3	aaaaaazbaaazbzbaaazbzbzbzbzb	0.0	100.0

Table 1: An example for which enforcing the constraints improves accuracy. Red indicates errors.

bzbzbzbzazbzbzazazazazbz⟶zbzbzbzbaaazbzbaaaaaaaaaaaaazb

iteration	output	loss	accuracy
0	zbzbzbzbaaazbaaaaaaaaaaazbaaa	0.2954	74.2
4	zbzbzbzbzbaaaaaaaaazbzbzbaaaaaa	0.0	60.0

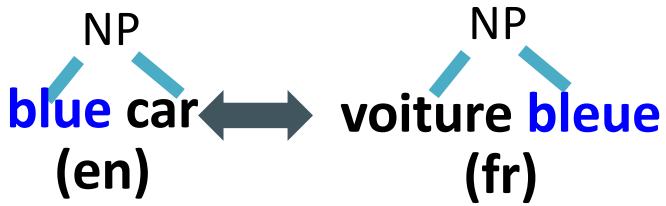
Table 2: An example for which enforcing the constraints degrades accuracy. Errors in red.

GBI on

Parsing slides

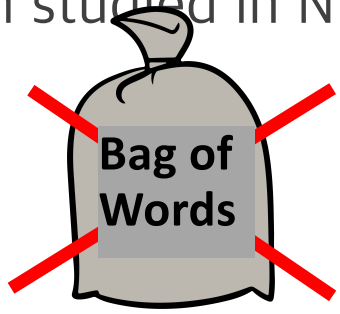
GBI on Syntactic Parsing

- Application 2: Syntactic constituency parsing.
- One of the fundamental problem studied in NLP for decades.
- Provides structure to NLP tasks.
  - Sentiment analysis, Translations, Named Entity Recognitions, ….



GBI on Syntactic Parsing

- Application 2: Syntactic constituency parsing.
- One of the fundamental problem studied in NLP for decades.
- Provides structure to NLP tasks.
  - Sentiment analysis, Translations, Named Entity Recognitions, ….
  - e.g.)
    “This film does not care about cleverness, wit or any other kind of intelligent humor"
    (Socher et al EMLNP 2013)



GBI on

GBI in detail

DETAIL

Learning with constraints

- **Proposed methods**
  - Output inconsistency (*OI-Loss*)

$$\text{SI-Loss} = -\, r(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}; \mathbf{w}^{(t)})$$

$$d(\mathbf{x}, \mathbf{y}) = \frac{|\text{disagreeing-spans}(\mathbf{x}, \mathbf{y})|}{|\text{srl-spans}(\mathbf{y})|}$$

$$r(\mathbf{x}, \mathbf{y}) = 1 - 2 \times d(\mathbf{x}, \mathbf{y})$$

Motivation

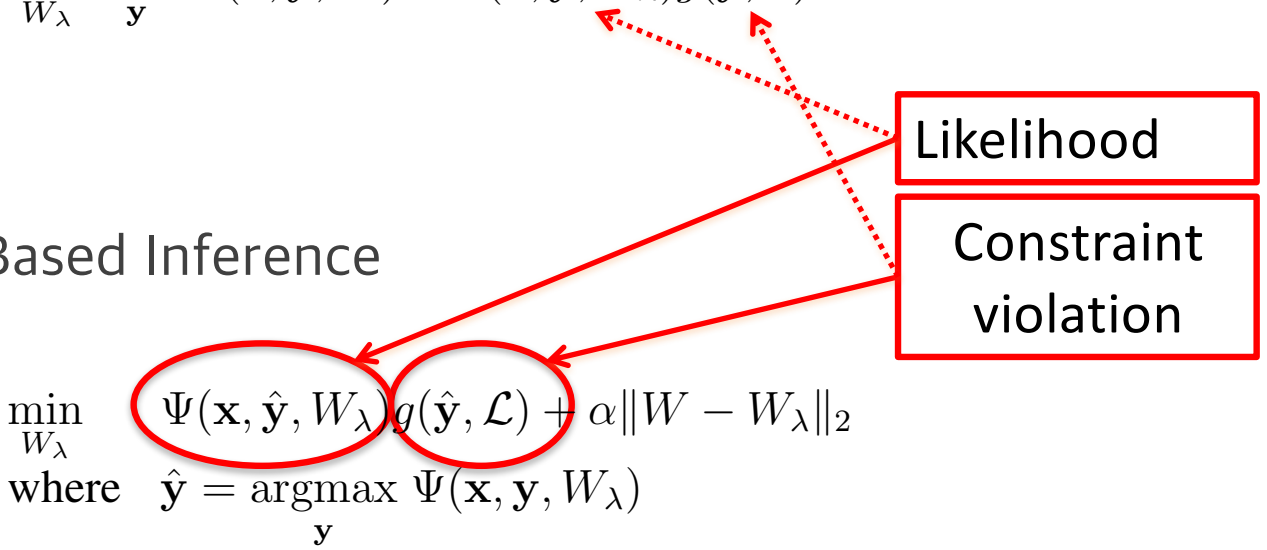
Gradient-Based Inference (GBI)

- Modified Lagrangian Relaxation

$$\min_{W_\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) g(\mathbf{y}, \mathcal{L})$$

- Gradient-Based Inference

$$\min_{W_\lambda} \quad \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$
$$\text{where} \quad \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$



Motivation

Gradient-Based Inference (GBI)

- Modified Lagrangian Relaxation

$$\min_{W_\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) g(\mathbf{y}, \mathcal{L})$$

- Gradient-Based Inference

$$\begin{aligned} \min_{W_\lambda} \quad & \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2 \\ \text{where} \quad & \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) \end{aligned}$$

Regularizer

Motivation

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\begin{aligned} \max_{\mathbf{y}} \quad & \Psi(\mathbf{x}, \mathbf{y}, W) \\ \text{s.t.} \quad & g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0 \end{aligned}$$

- Standard Lagrangian Relaxation

$$\min_{\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \lambda g(\mathbf{y}, \mathcal{L})$$

Motivation

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\begin{aligned} \max_{\mathbf{y}} \quad & \Psi(\mathbf{x}, \mathbf{y}, W) \\ \text{s.t.} \quad & g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0 \end{aligned}$$

- Standard Lagrangian Relaxation

$$\min_{\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \lambda g(\mathbf{y}, \mathcal{L})$$

- Modified Lagrangian Relaxation

$$\min_{W_\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) g(\mathbf{y}, \mathcal{L})$$

Likelihood

Constraint violation

Motivation

Gradient-Based Inference (GBI)

- Standard Lagrangian Relaxation

$$\min_{\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \lambda g(\mathbf{y}, \mathcal{L})$$

- Gradient-Based Inference

$$\begin{aligned} \min_{W_\lambda} \quad & \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2 \\ \text{where} \quad & \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) \end{aligned}$$

Motivation

Gradient-Based Inference (GBI)

- Modified Lagrangian Relaxation

$$\min_{W_\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) g(\mathbf{y}, \mathcal{L})$$

- Gradient-Based Inference

$$\begin{aligned} \min_{W_\lambda} \quad & \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2 \\ \text{where} \quad & \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) \end{aligned}$$

Motivation

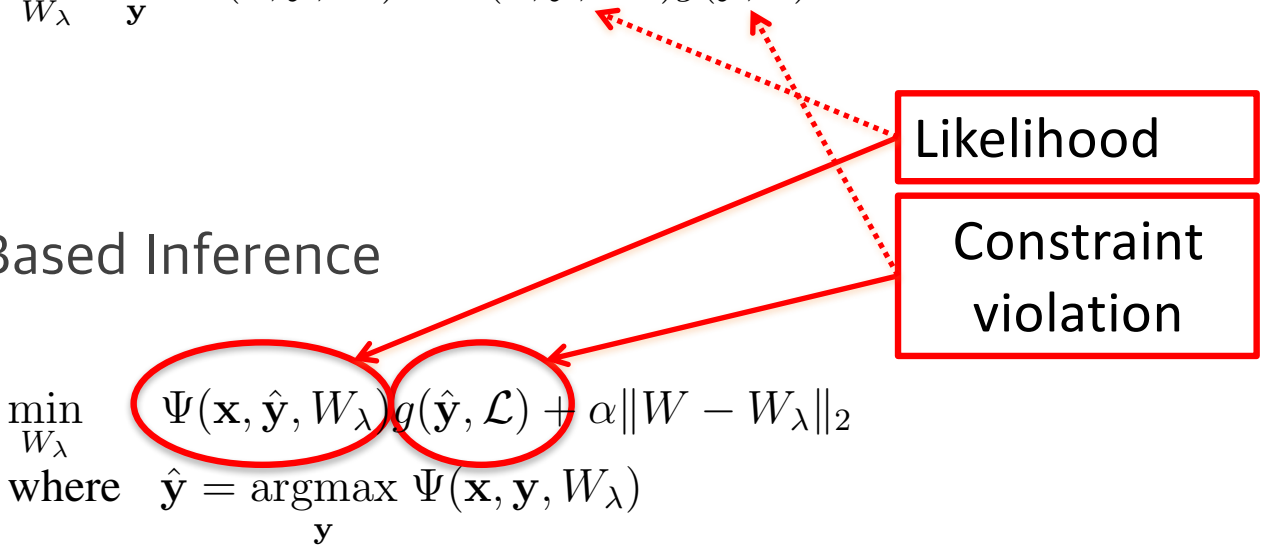
Gradient-Based Inference (GBI)

- Modified Lagrangian Relaxation

$$\min_{W_\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) g(\mathbf{y}, \mathcal{L})$$

- Gradient-Based Inference

$$\min_{W_\lambda} \quad \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$
$$\text{where} \quad \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$



Motivation

Gradient-Based Inference (GBI)

- Modified Lagrangian Relaxation

$$\min_{W_\lambda} \max_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W) + \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) g(\mathbf{y}, \mathcal{L})$$

- Gradient-Based Inference

$$\begin{aligned} \min_{W_\lambda} \quad & \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2 \\ \text{where} \quad & \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) \end{aligned}$$

Regularizer

Gradient-Based Inference (GBI)

- Optimization with constraints

$$\begin{aligned} \max_{\mathbf{y}} \quad & \Psi(\mathbf{x}, \mathbf{y}, W) \\ \text{s.t.} \quad & g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) = 0 \end{aligned}$$

- Gradient-Based Inference (in math)

$$\begin{aligned} \min_{W_\lambda} \quad & \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2 \\ \text{where} \quad & \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) \end{aligned}$$

DETAIL

Gradient-Based Inference (GBI)

- GBI (in Algorithm)

$$\min_{W_\lambda} \quad \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$
$$\text{where} \quad \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$

Algorithm 1 Constrained inference for neural nets

Inputs: test instance $\mathbf{x}$, input specific CFL $\mathcal{L}^{\mathbf{x}}$, pretrained weights W

$W_\lambda \leftarrow W$ #reset instance-specific weights

Initiate as model weight

while not converged **do**

$\mathbf{y} \leftarrow f(\mathbf{x}; W_\lambda)$ #perform inference using weights W_λ

$\nabla \leftarrow g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) \frac{\partial}{\partial W_\lambda} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) + \alpha \frac{W - W_\lambda}{\|W - W_\lambda\|_2}$ #compute constraint loss

$W_\lambda \leftarrow W_\lambda - \eta \nabla$ #update instance-specific weights with SGD or a variant thereof

end while

Jay Yoon Lee et al, Gradient-based Inference for Networks with Output Constraint, AAAI 2019

DETAIL

Gradient-Based Inference (GBI)

- GBI (in Algorithm)

$$\min_{W_\lambda} \quad \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$

$$\text{where} \quad \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$

Algorithm 1 Constrained inference for neural nets

Inputs: test instance $\mathbf{x}$, input specific CFL $\mathcal{L}^{\mathbf{x}}$, pretrained weights W

$W_\lambda \leftarrow W$ #reset instance-specific weights

while not converged **do**

$\mathbf{y} \leftarrow f(\mathbf{x}; W_\lambda)$ #perform inference using weights W_λ

$\nabla \leftarrow g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) \frac{\partial}{\partial W_\lambda} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) + \alpha \frac{W - W_\lambda}{\|W - W_\lambda\|_2}$ #compute constraint loss

$W_\lambda \leftarrow W_\lambda - \eta \nabla$ #update instance-specific weights with SGD or a variant thereof

end while

Jay Yoon Lee et al, Gradient-based Inference for Networks with Output Constraint, AAAI 2019

DETAIL

Gradient-Based Inference (GBI)

- GBI (in Algorithm)

$$\min_{W_\lambda} \quad \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$
$$\text{where} \quad \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$

Algorithm 1 Constrained inference for neural nets

Inputs: test instance $\mathbf{x}$, input specific CFL $\mathcal{L}^{\mathbf{x}}$, pretrained weights W
$W_\lambda \leftarrow W$ #reset instance-specific weights
while not converged **do**
  $\mathbf{y} \leftarrow f(\mathbf{x}; W_\lambda)$ #perform inference using weights W_λ
  $\nabla \leftarrow g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) \frac{\partial}{\partial W_\lambda} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) + \alpha \frac{W - W_\lambda}{\|W - W_\lambda\|_2}$ #compute constraint loss
  $W_\lambda \leftarrow W_\lambda - \eta \nabla$ #update instance-specific weights with SGD or a variant thereof
end while

Jay Yoon Lee et al, Gradient-based Inference for Networks with Output Constraint, AAAI 2019

DETAIL

Gradient-Based Inference (GBI)

- GBI (in Algorithm)

$$\min_{W_\lambda} \quad \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) g(\hat{\mathbf{y}}, \mathcal{L}) + \alpha \|W - W_\lambda\|_2$$
$$\text{where} \quad \hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda)$$

Algorithm 1 Constrained inference for neural nets

Inputs: test instance $\mathbf{x}$, input specific CFL $\mathcal{L}^{\mathbf{x}}$, pretrained weights W

$W_\lambda \leftarrow W$ #reset instance-specific weights

while not converged **do**

$\mathbf{y} \leftarrow f(\mathbf{x}; W_\lambda)$ #perform inference using weights W_λ

$\nabla \leftarrow g(\mathbf{y}, \mathcal{L}^{\mathbf{x}}) \frac{\partial}{\partial W_\lambda} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) + \alpha \frac{W - W_\lambda}{\|W - W_\lambda\|_2}$ #compute constraint loss

$W_\lambda \leftarrow W_\lambda - \eta \nabla$ #update instance-specific weights with SGD or a variant thereof

end while

Reset for new instance

Jay Yoon Lee et al, Gradient-based Inference for Networks with Output Constraint, AAAI 2019

Ganchev et al., 2010, Hu et al., (ACL2016, EMNLP2016, NIPS2018), Hinton et al, 2016

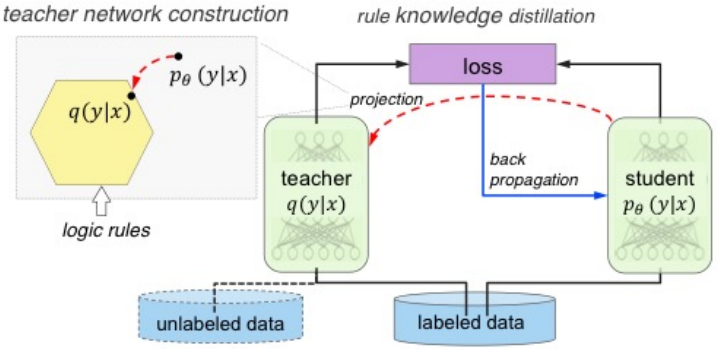
Posterior Regularization & Distillation

- Using posterior regularization,

$$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N}(1-\pi)\ell(\mathbf{y}_n, \boldsymbol{\sigma}_\theta(\mathbf{x}_n)) + \pi\ell(\mathbf{s}_n^{(t)}, \boldsymbol{\sigma}_\theta(\mathbf{x}_n))$$

sampling

$$q^*(\mathbf{Y}|\mathbf{X}) \propto p_\theta(\mathbf{Y}|\mathbf{X}) \exp\left\{-\sum_{l,g_l} C\lambda_l(1 - r_{l,g_l}(\mathbf{X},\mathbf{Y}))\right\}$$



- The base distribution is only stable for problems that current output does not depend on previous output (no seq2seq). (Hu 2016)

- Different way of injecting score. (Mine is direct score of rules whereas PR samples from base probability times exponentiated payoff function.

- Also, sampling (Hu 2018) from q distrbution is required, not easy to do for complicated structures.
  - The NIPS2018 paper does importance sampling with MC sampling on normalization factor. (The author told me that he only samples 5 times)
  - The distribution changes constantly, and not sure how well 5 step MC step will approximate normalization factor.