

Injecting constraints into neural NLP models

Jay-Yoon Lee

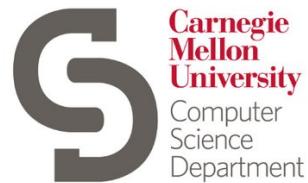
Thesis committee

Jaime Carbonell (Chair)

William Cohen

Graham Neubig

Dan Roth, University of Pennsylvania



Language
Technologies
Institute



**Carnegie
Mellon
University**

Injecting constraints into neural NLP models

Constraints?

Injecting **constraints**
into neural NLP models.

Constraints?

- **Grammar**

- 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

- 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)

Constraints?

- Image-captioning

**Input
(image)**



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

Constraints?

- Image-captioning

Input
(image)



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.

Constraint violating

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

Constraints?

- Image-captioning

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.

Injecting constraints into neural NLP models

neural NLP

Injecting constraints
into **neural NLP** models.
(Natural Language Processing)

neural NLP



Machine
Translation
(automatic)

(Input)

English: Have a nice day.

(output)

Korean: 좋은 하루 되세요!



Question
&
Answering

(Input)

Sentence: John broke a window with a hammer.

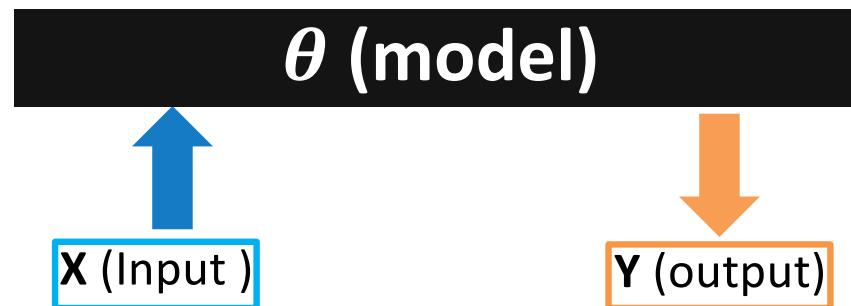
Question: What did John use to break the window?

(output)

Answer: Hammer

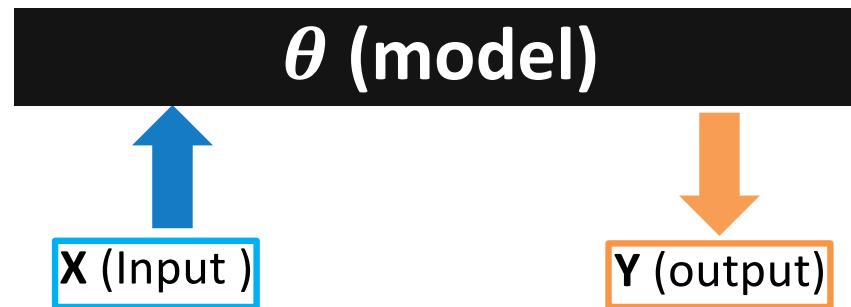
Typical neural model training

- Black box trained with cross-entropy loss
 - Maximizing log-likelihood of (input, true output) pair.
 - Often requires massive dataset.



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 known constraints?*



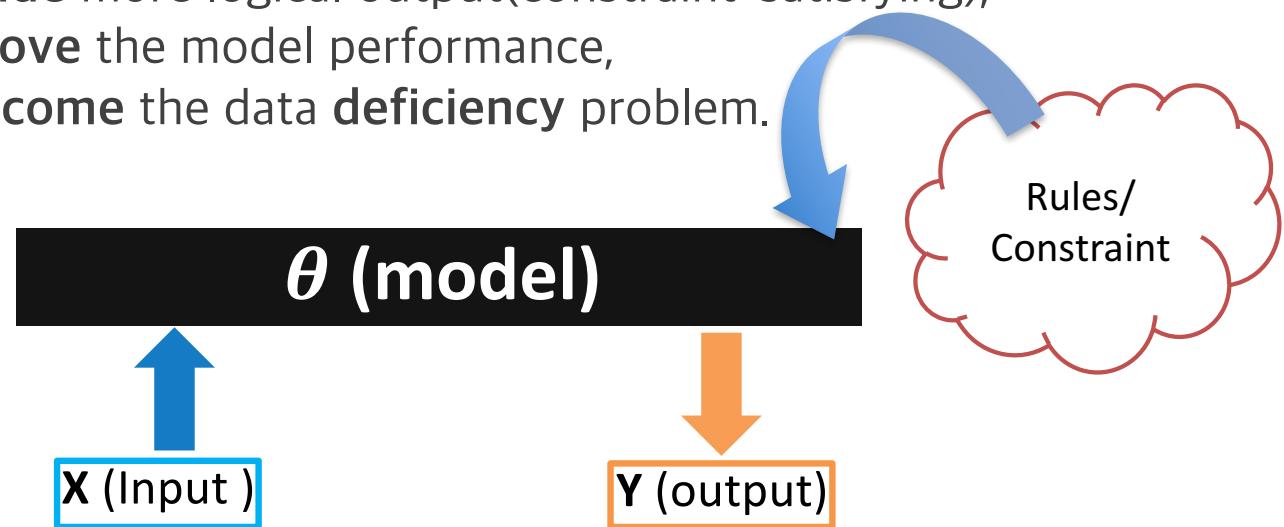
Challenges in injecting constraints

- 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,
 - Overcome the data **deficiency** problem.



Roadmap

- **Applications & constraints**

- Inference with constraints [AAAI 19']
- Learning with constraints [EMNLP 18']

Completed
Work

- Proposed Work
 - New methods (exploration).
 - New application.

Considered Application

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

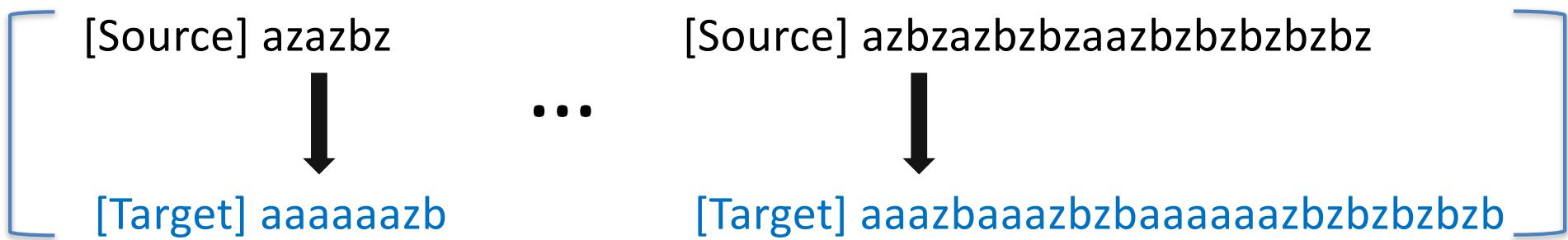
$$T : \mathcal{L}_S \rightarrow \mathcal{L}_T$$

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



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 \rightarrow aaa
bz \rightarrow zb

S: azazbzazbzazbzbzbzbz

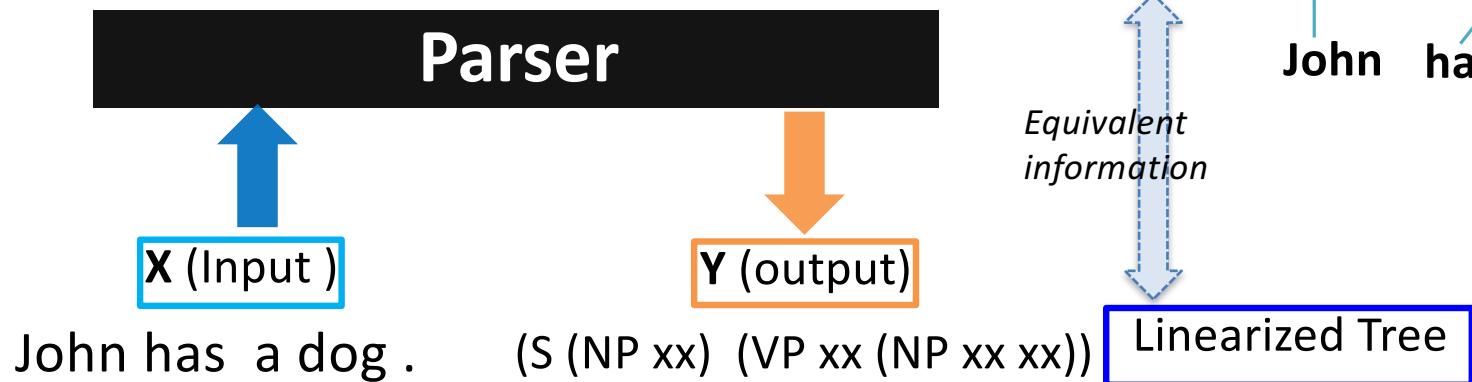


constraint:
 $3 \times (\# \text{ a in } S) = \# \text{ a in } T$

T: aaaaaazbaaabzbzaazbzbzbz

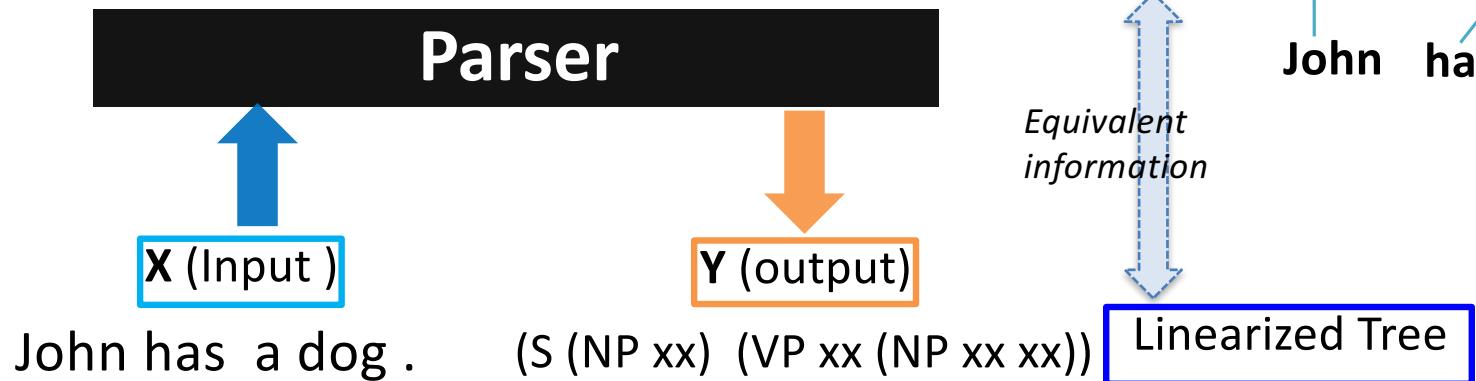
Considered Applications in NLP

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



Considered Applications in NLP

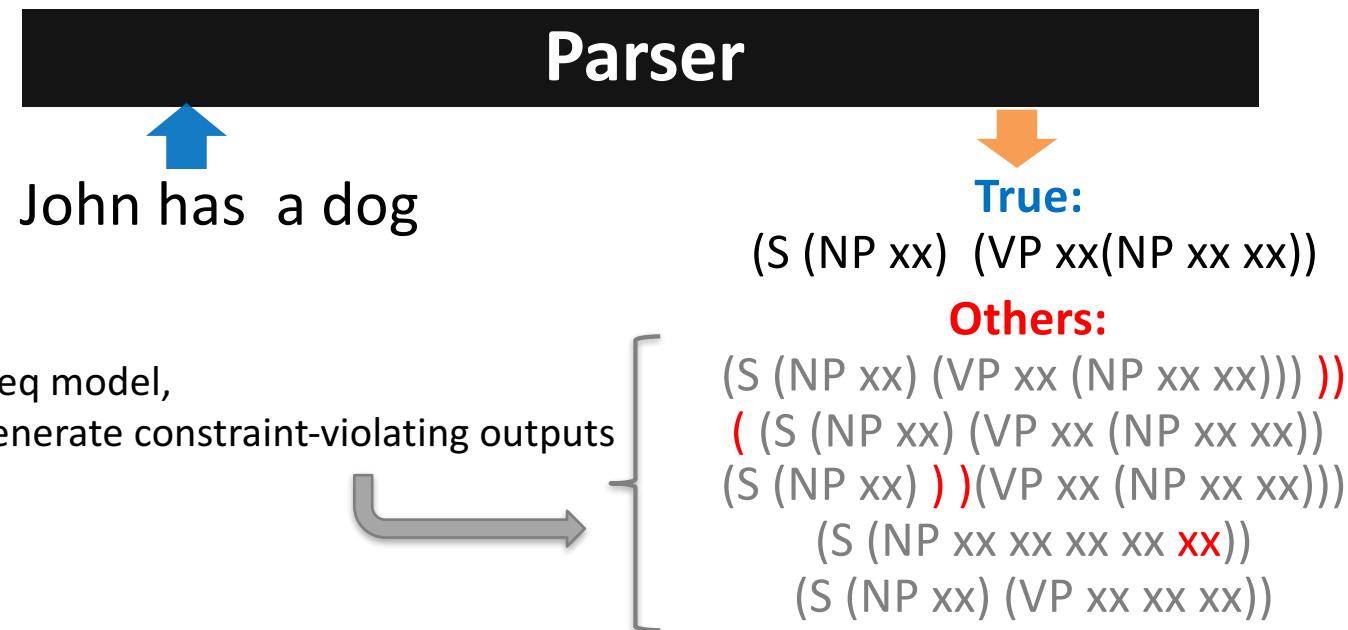
- *Syntactic Parsing*
 - Finding structure of the provided (input) sentence.



To use *Sequence-to-Sequence* model:
models relationship between *sequence* input & *sequence* output.

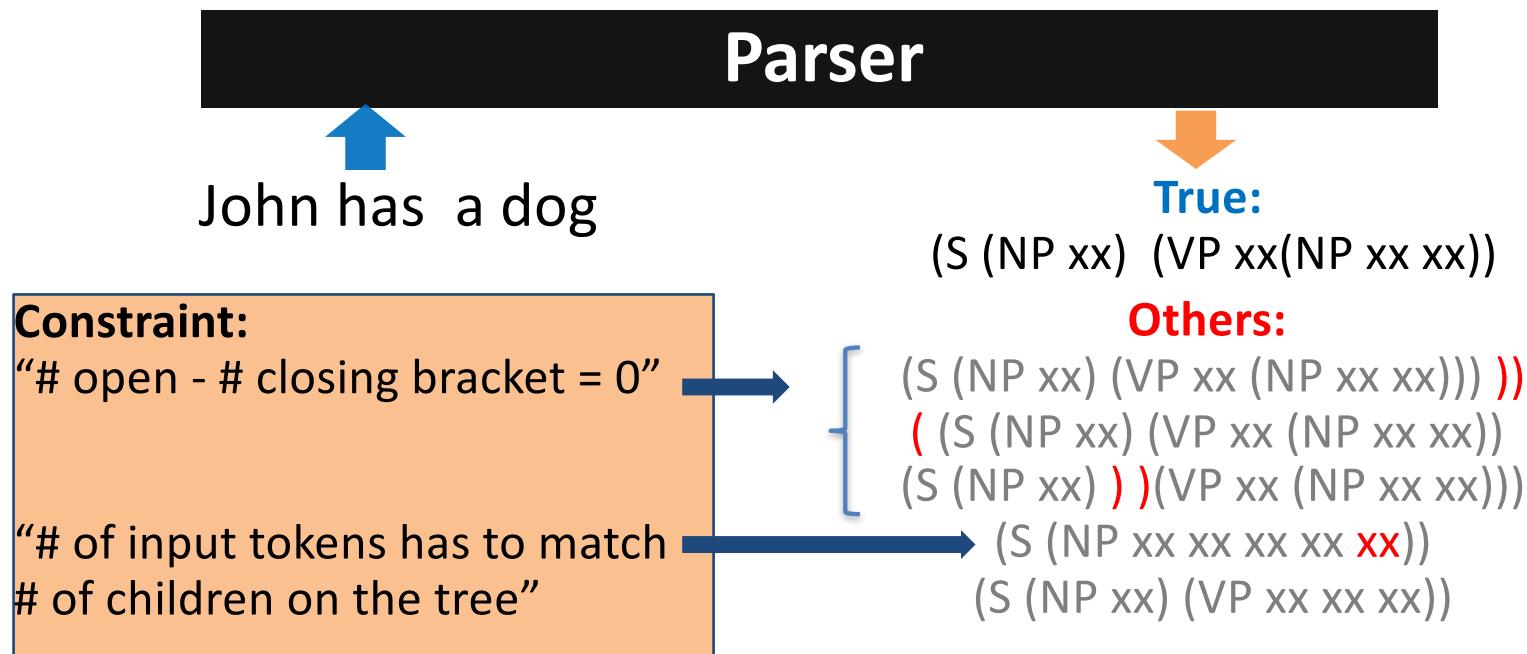
Constraints in Parsing

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



Constraints in Parsing

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



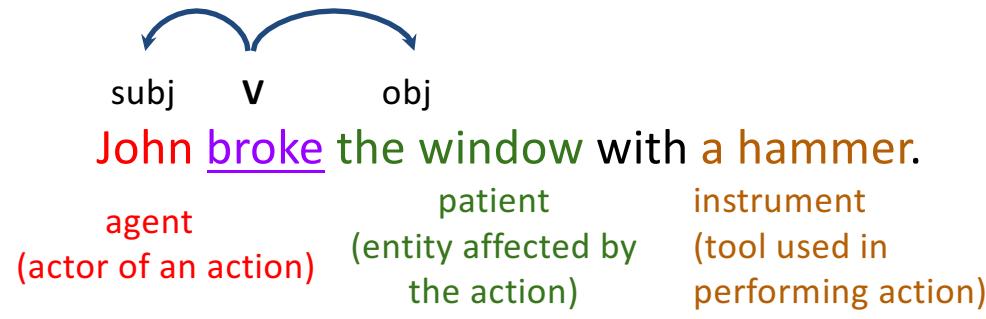
Considered Applications in NLP

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



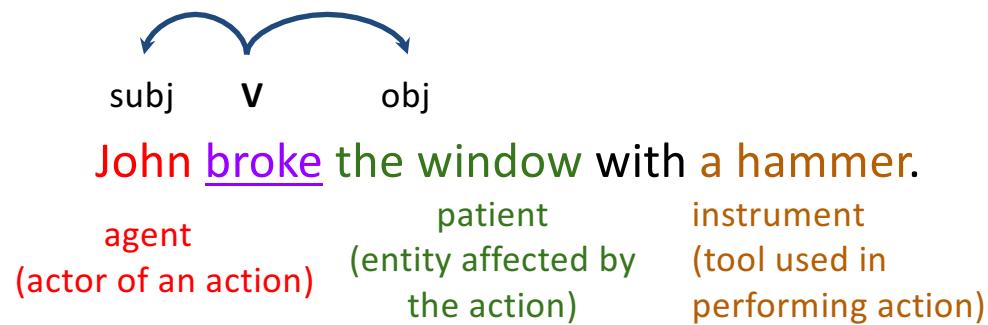
Semantic Role Labeling

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



Semantic Role Labeling

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



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

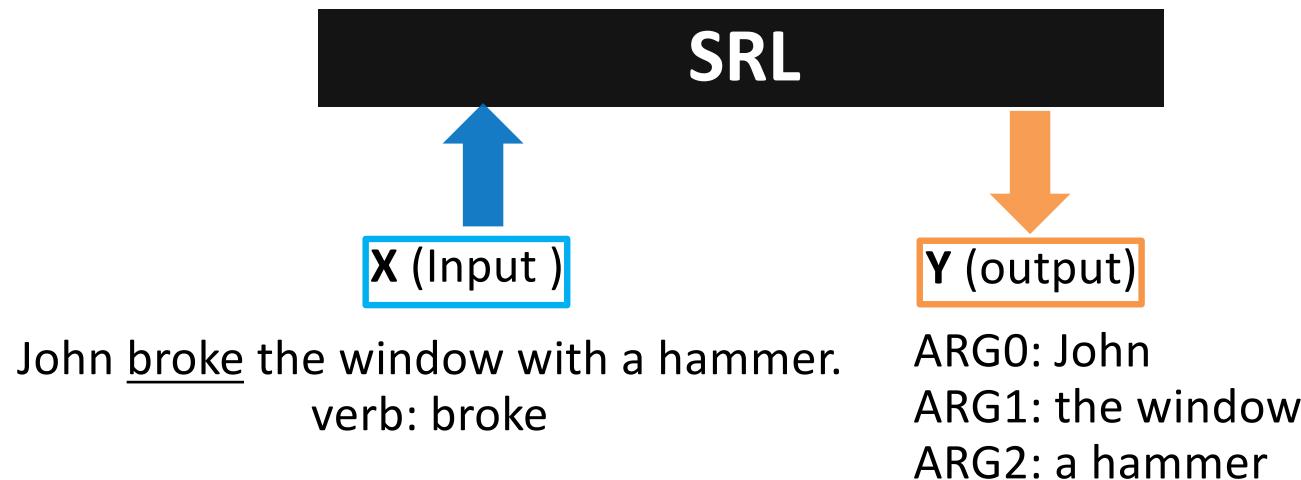
Role	Description
ARGM-TMP	temporal
ARGM-DIR	direction
ARGM-LOC	location
...	...

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.**

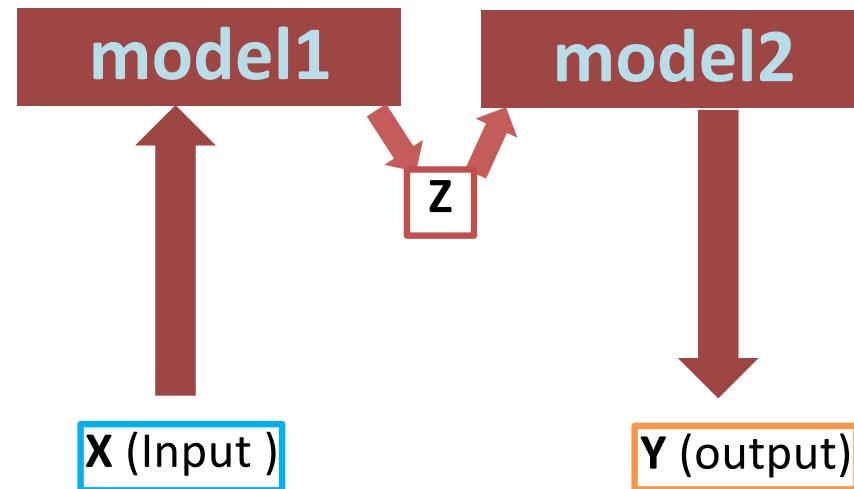
Semantic Role Labeling

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



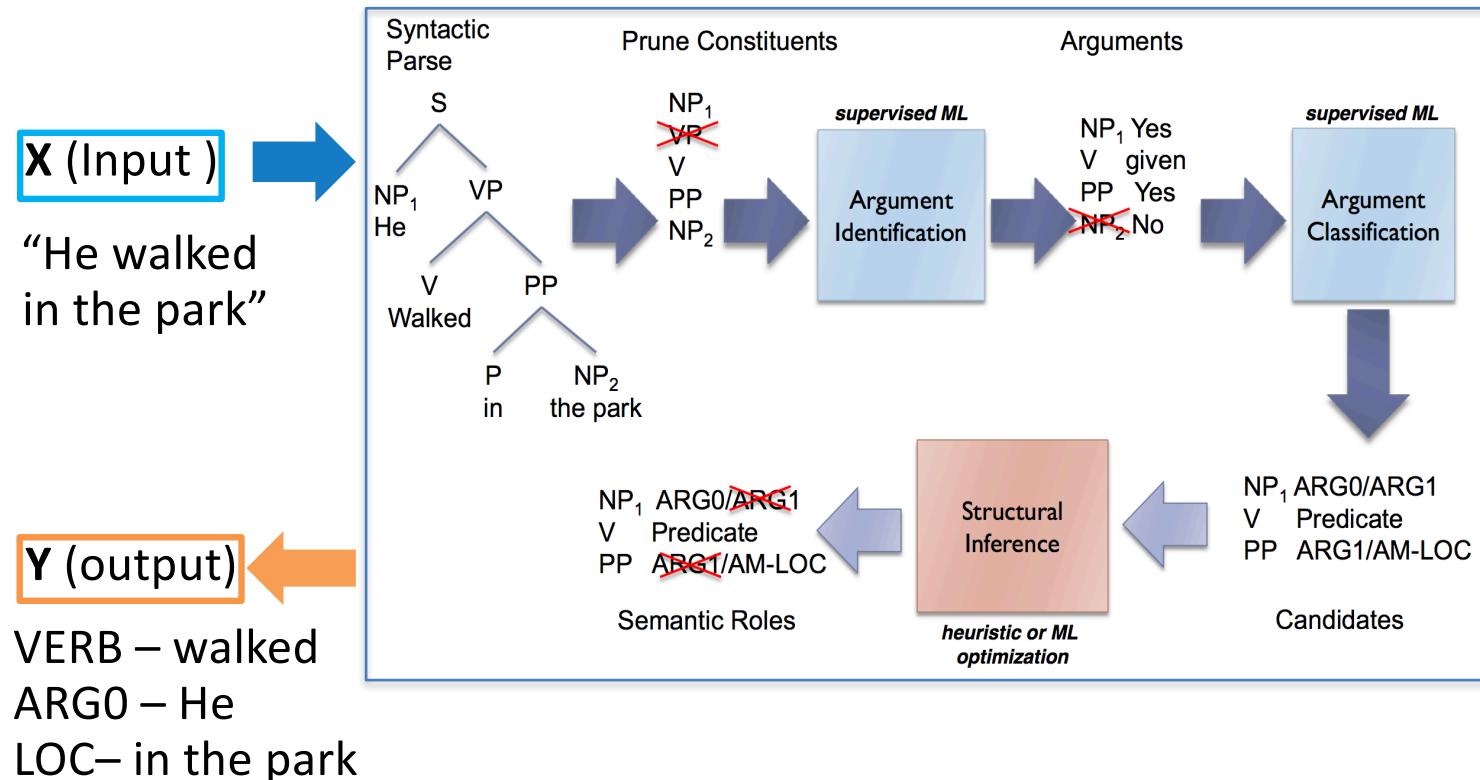
Complex task

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



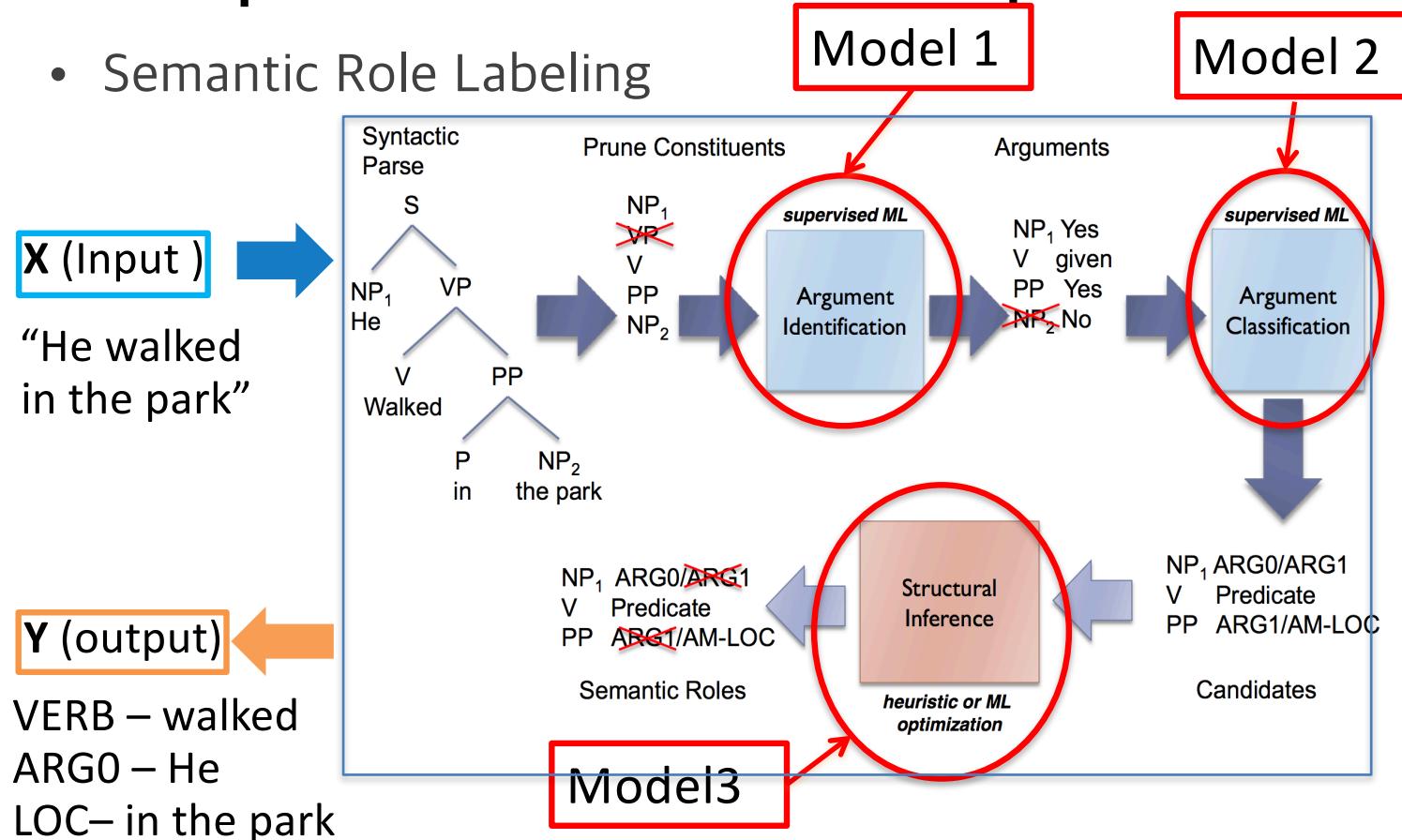
Complex models - example

- Semantic Role Labeling (step by step)



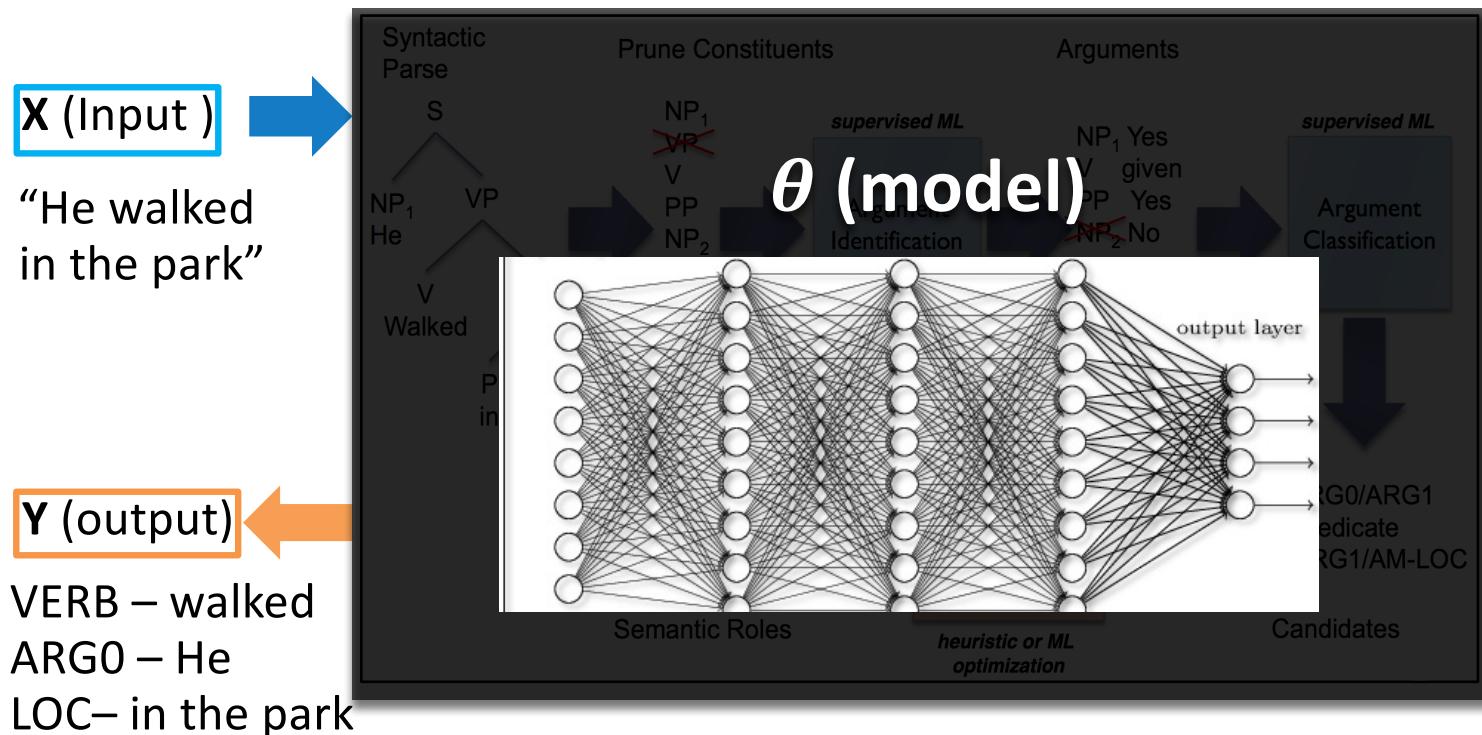
Complex models - example

- 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.



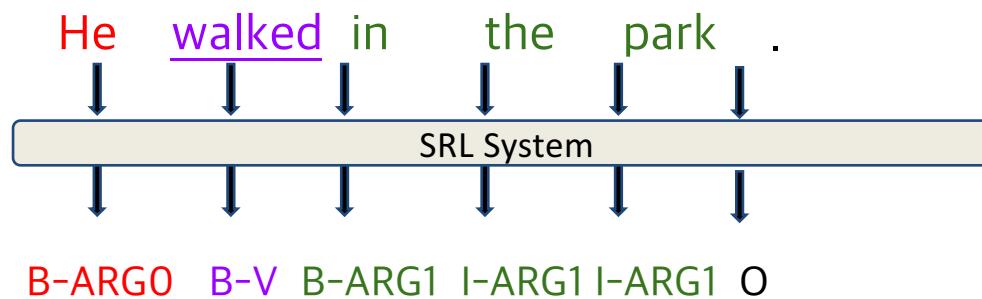
How to solve it?

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:



What are different constraints for SRL task?

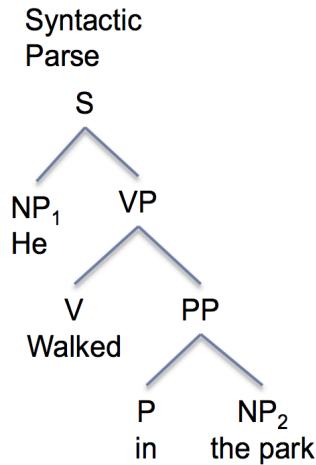
1. BIO constraints
 - Constraint violation: B-ARG0 followed by I-ARG1
 - (similar to the constraints in Parsing example)
2. SRL constraints (as defined by Punyakanok et al. (2008))
 - Unique core roles (U)
 - Continuation roles (C)
 - Reference roles (R)
3. Syntactic constraints
 - Constraint violation: Arguments that are not constituents within a given parse tree

Syntactic constraint on SRL

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

X (Input)

“He walked
in the park”



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



Possible span candidate

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

(He) (walked) (in the park)

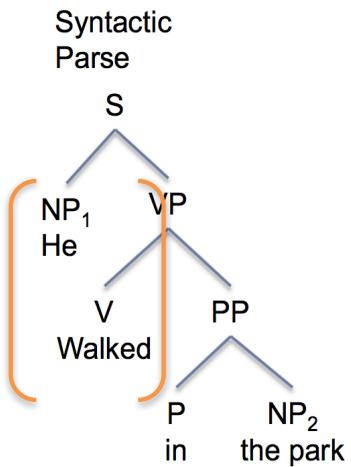
(He) (walked in)(the park)
(He walked) (in the park)
(He walked in the)(park)
(He walked in the park)

Syntactic constraint on SRL

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

X (Input)

“He walked
in the park”



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



Possible span candidate

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

(He) (walked) (in the park)

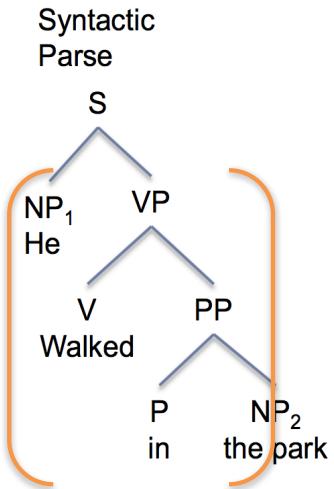
(He) (walked in)(the park)
(He walked) (in the park)
(He walked in the)(park)
(He walked in the park)

Syntactic constraint on SRL

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

X (Input)

“He walked
in the park”



However, end-to-end model without
syntactic information can generate

Constraint-violating “spans” given sentence.



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)

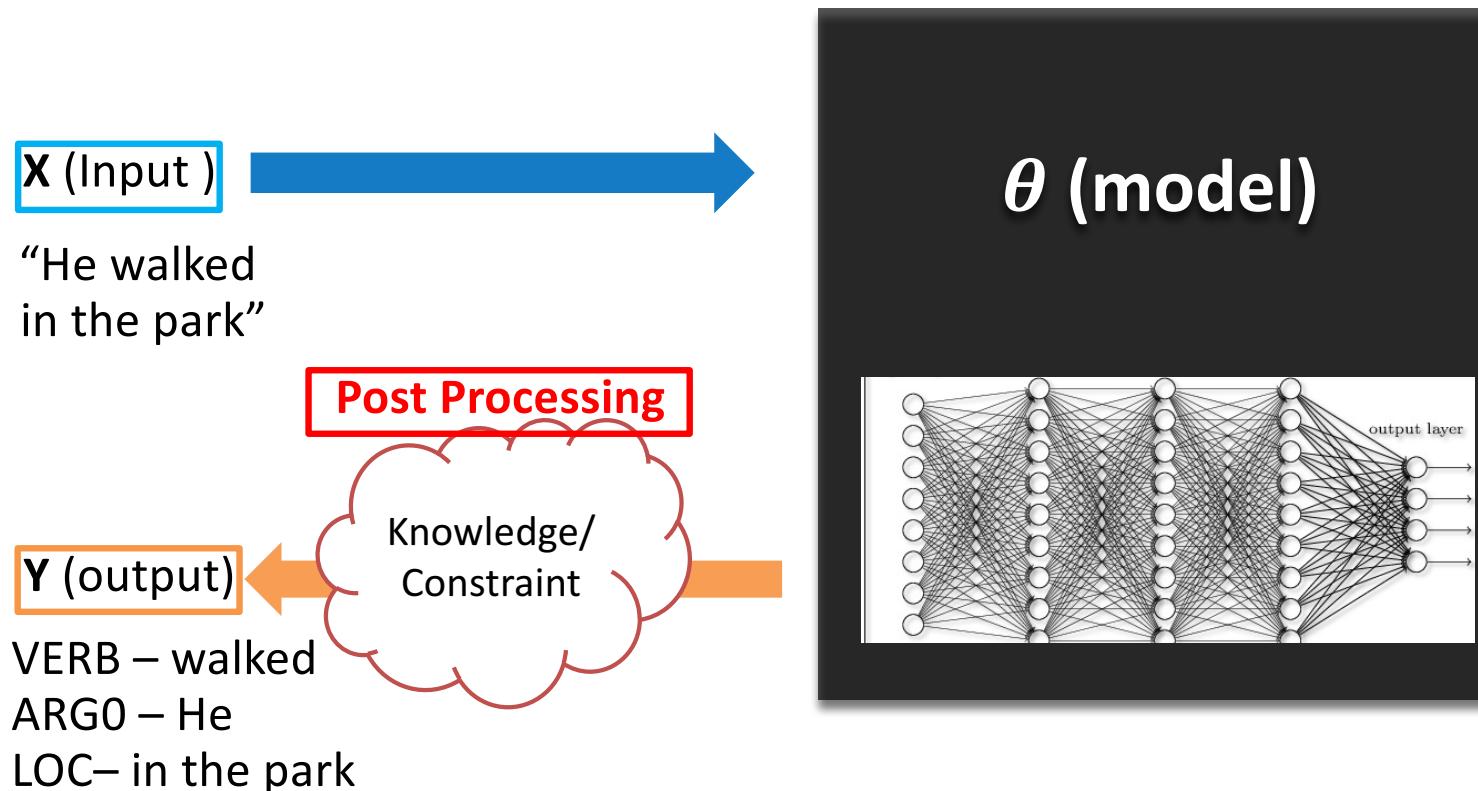
(He walked) (in the park)

(He walked in the)(park)

(He walked in the park)

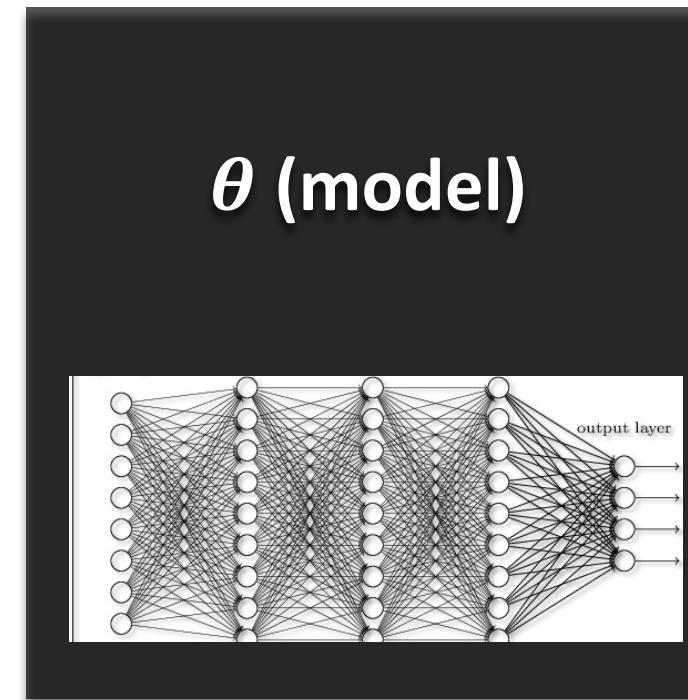
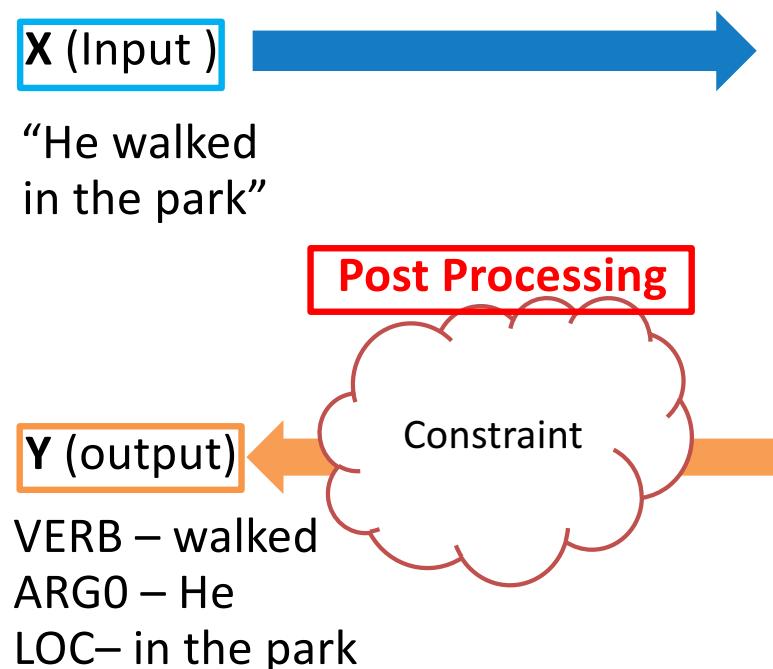
Inference using constraint

- Inference: Finding the best output given the model.



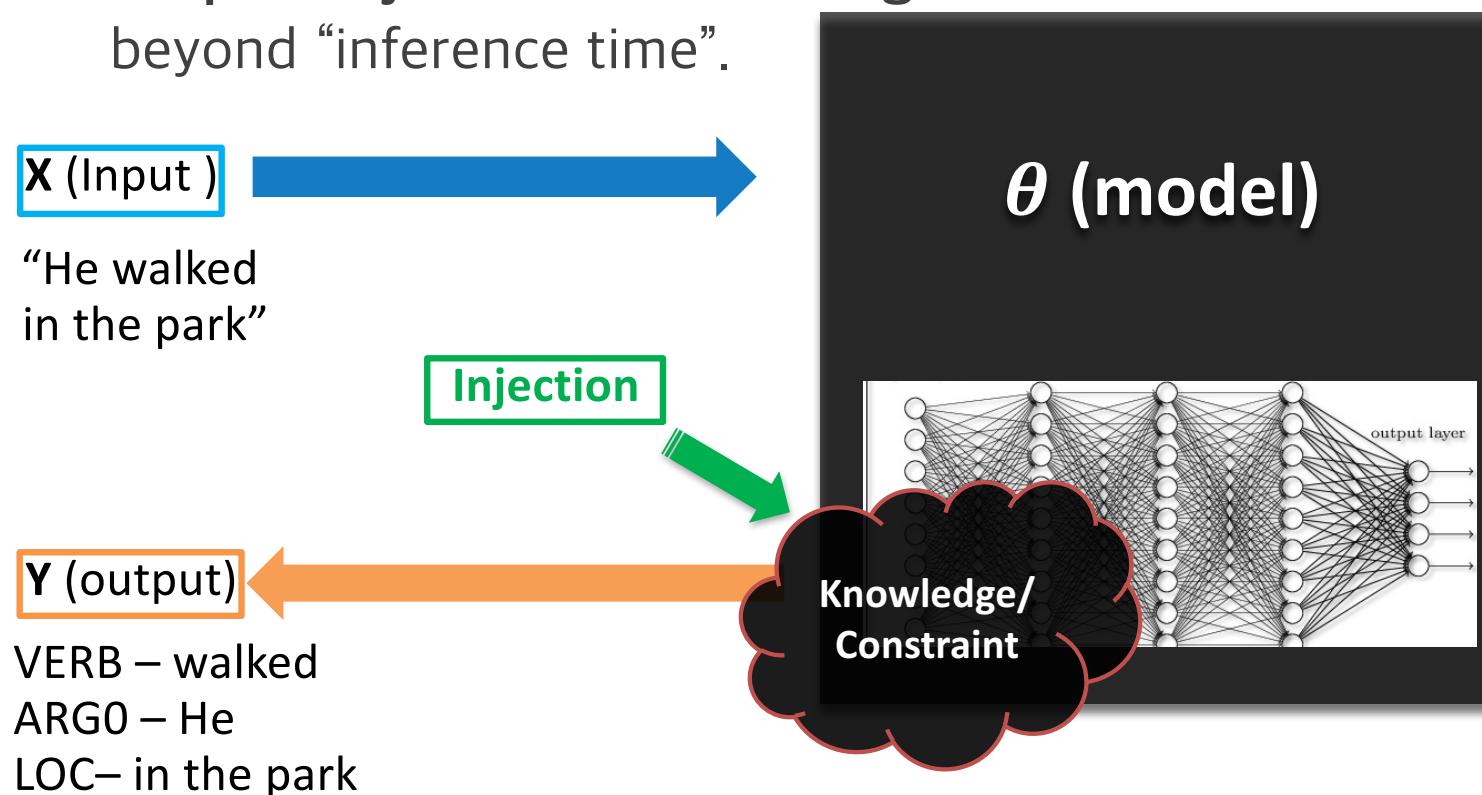
Neural Models – Inference

- Proper “Injection of knowledge” on Inference step.



Learning with constraints

- Proper “Injection of knowledge” to the model beyond “inference time”.



Roadmap

- Applications & constraints

- **Inference with constraints [AAAI 19']**

- Learning with constraints [EMNLP 18']

- Proposed Work
 - New methods (exploration).
 - New application.

Completed
Work

Gradient-Based Inference* (GBI)

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

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


A mathematical expression showing the optimization of a function Ψ with respect to \mathbf{y} , given inputs \mathbf{x} and parameters W . A green circle highlights the parameter W , and a green arrow points from the text "Learned model" to this circle.

Gradient-Based Inference* (GBI)

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

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

- 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

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

Constraint function

- Constraint (evaluation) function. $g(\mathbf{y}, \mathcal{L}) \rightarrow \mathbb{R}_0^+$
(Error counting)
- 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

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



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

Constraint
for feasible set $\mathcal{L}^{\mathbf{x}}$

Gradient-Based Inference* (GBI)

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

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

constraint

- 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

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

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

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

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 not suitable for the 'global' constraint. (not localizable)

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

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?

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

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

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

Likelihood

Constraint
violation

Motivation

Gradient-Based Inference (GBI)

- Optimization with constraints

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

Learned model weight

- Gradient-Based Inference

$$\begin{aligned} \min_{W_\lambda} \quad & \Psi(\mathbf{x}, \hat{\mathbf{y}}, W_\lambda) \\ \text{where} \quad & \hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}, W_\lambda) \end{aligned}$$

Expected constraint violation

Regularizer

Motivation

Gradient-Based Inference (GBI)

- Optimization with constraints Standard Lagrangian

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

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

Very slow. (combinatorial)
Especially global constraint

- Gradient-Based Inference (in math)

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

Greedy, beam-search
relatively fast

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

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

Combinatorial problem.

Very slow. (combinatorial)
Especially global constraint

- Gradient-Based Inference (in math)

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

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

Greedy, beam-search
relatively fast

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

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

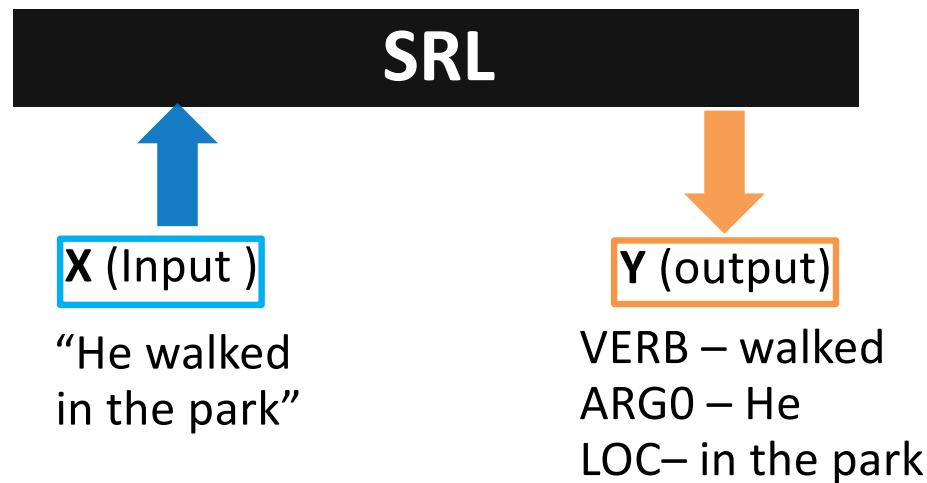
- Q1. Does GBI actually *enforce constraints?*
- Q2. Does GBI enforce constraints
without compromising the *F1 score* of the system?
- Q3. *Can GBI generalize* over multiple baseline model?
- Q4. Performance of GBI on out-of-domain dataset?

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. Performance of *GBI on out-of-domain* dataset?

Back to SRL

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

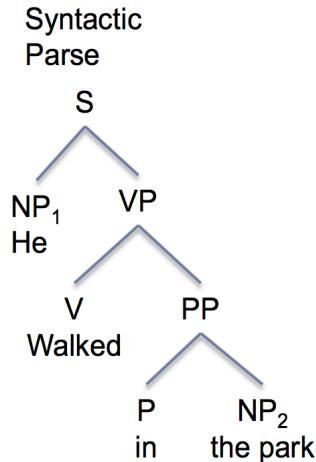


Syntactic constraint on SRL

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

X (Input)

“He walked
in the park”



Wrong spans marked with strike



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)

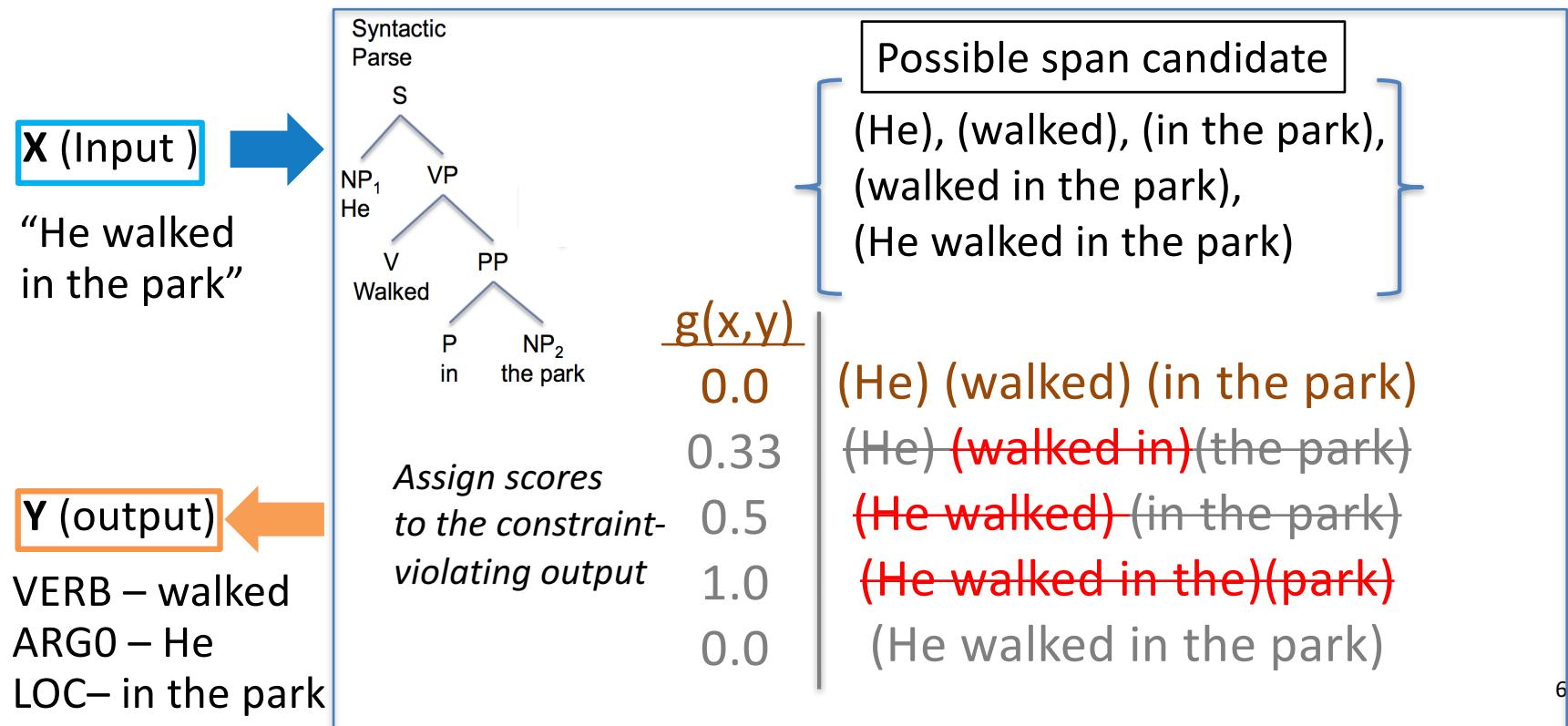
Constraint on SRL

- $\text{srl-spans}(\mathbf{y})$: spans of predicted SRL output \mathbf{y}
- $\text{parse-spans}(\mathbf{x})$: parsing constituents without labels.
- $\text{disagreeing-spans}(\mathbf{x}, \mathbf{y}) = \{span_i \in \text{srl-spans}(\mathbf{y}) \mid span_i \notin \text{parse-spans}(\mathbf{x})\}$
- 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 example



GBI on SRL: Experiments

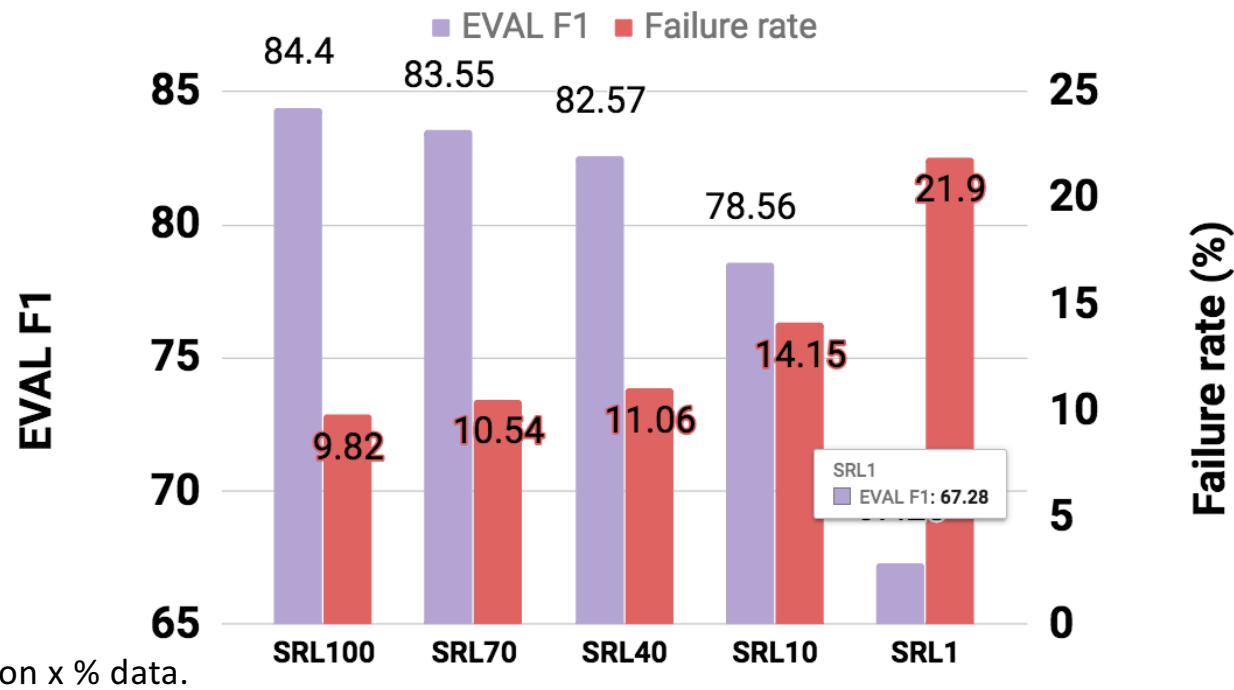
- OntoNotes v5.0, CoNLL-2012 shared task
(278k train, 38.3k dev, 25.6k test)
- Gold predicates and gold parse constituents are given
(~10% noisy annotations) .
- Baseline architecture
 - multi-layer highway bi-directional LSTM (ELMo embedding)

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.

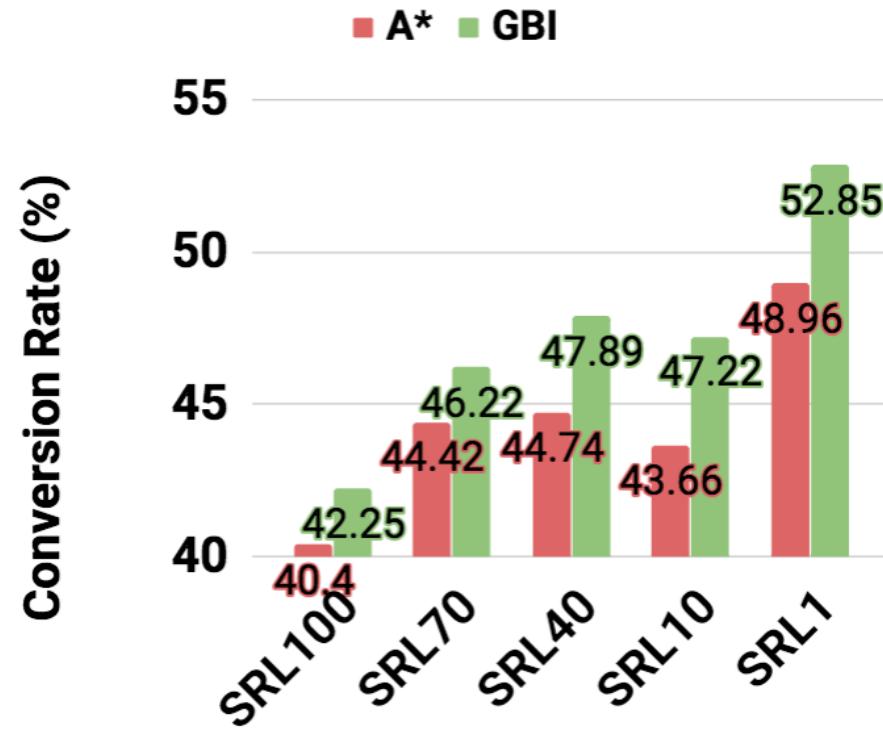
GBI on SRL: Experiments

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



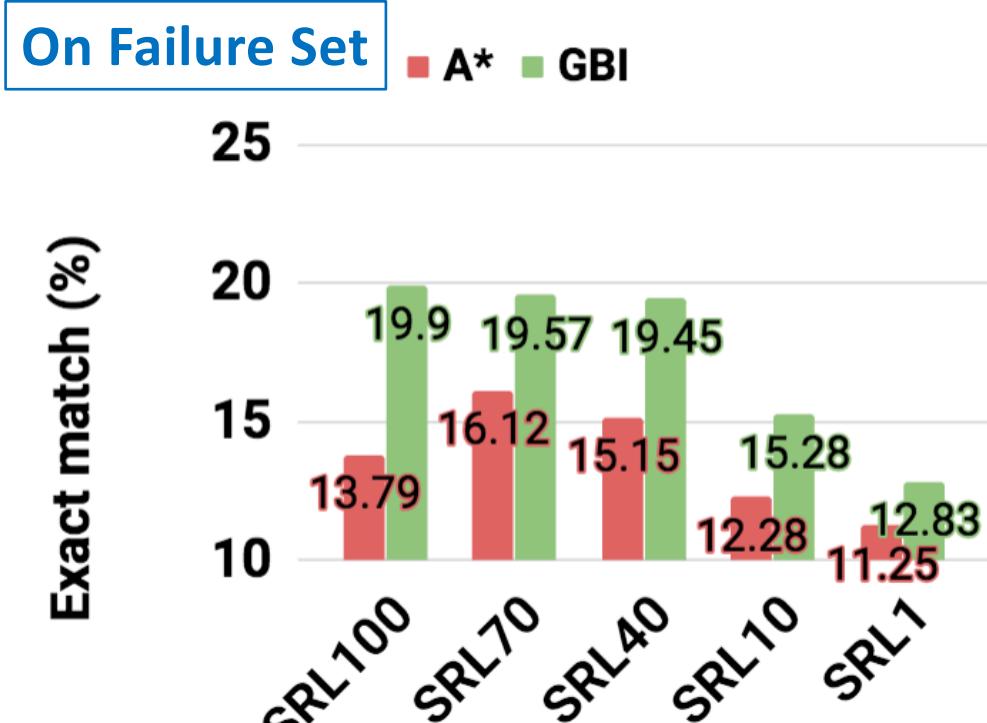
GBI on SRL: Experiments

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



GBI on SRL: Experiments

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

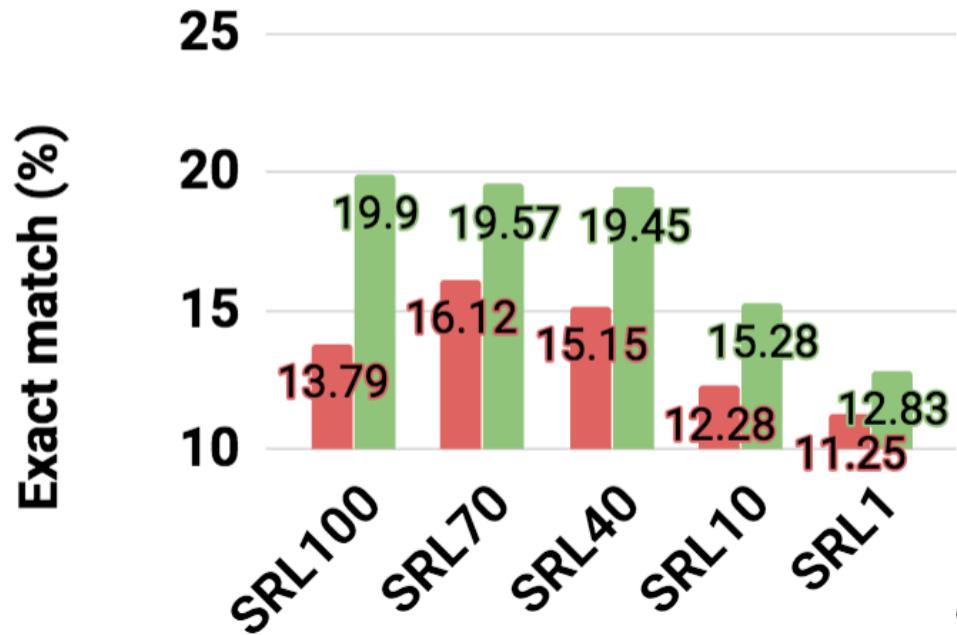


SRL x : SRL system trained on x % data.

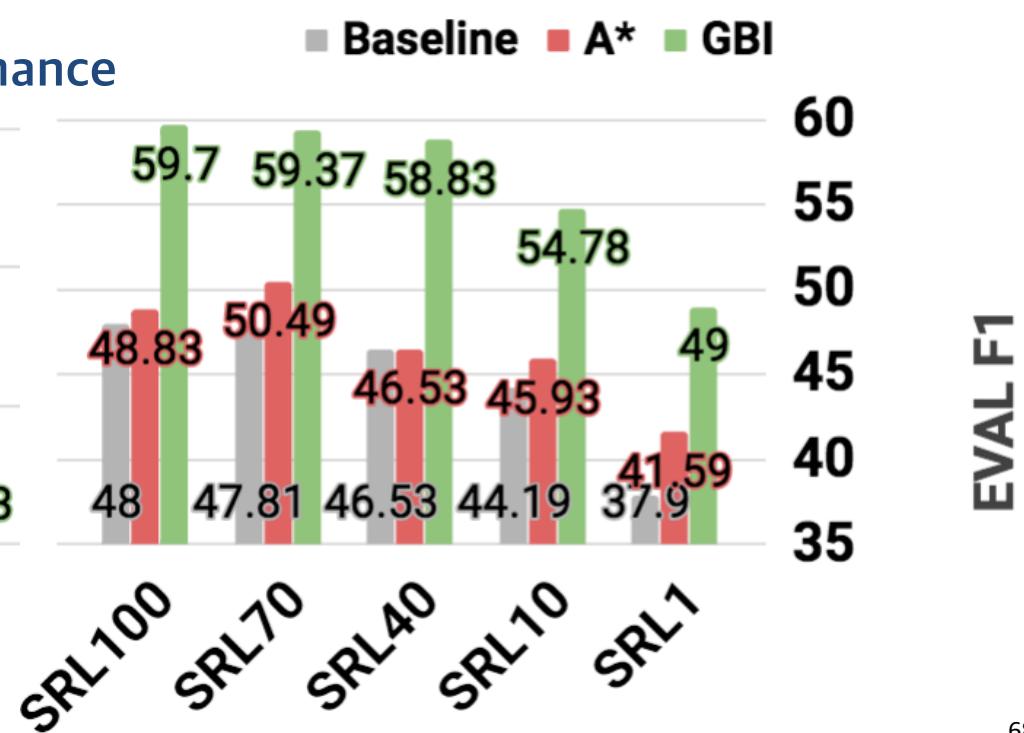
GBI on SRL: Experiments

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

On Failure Set



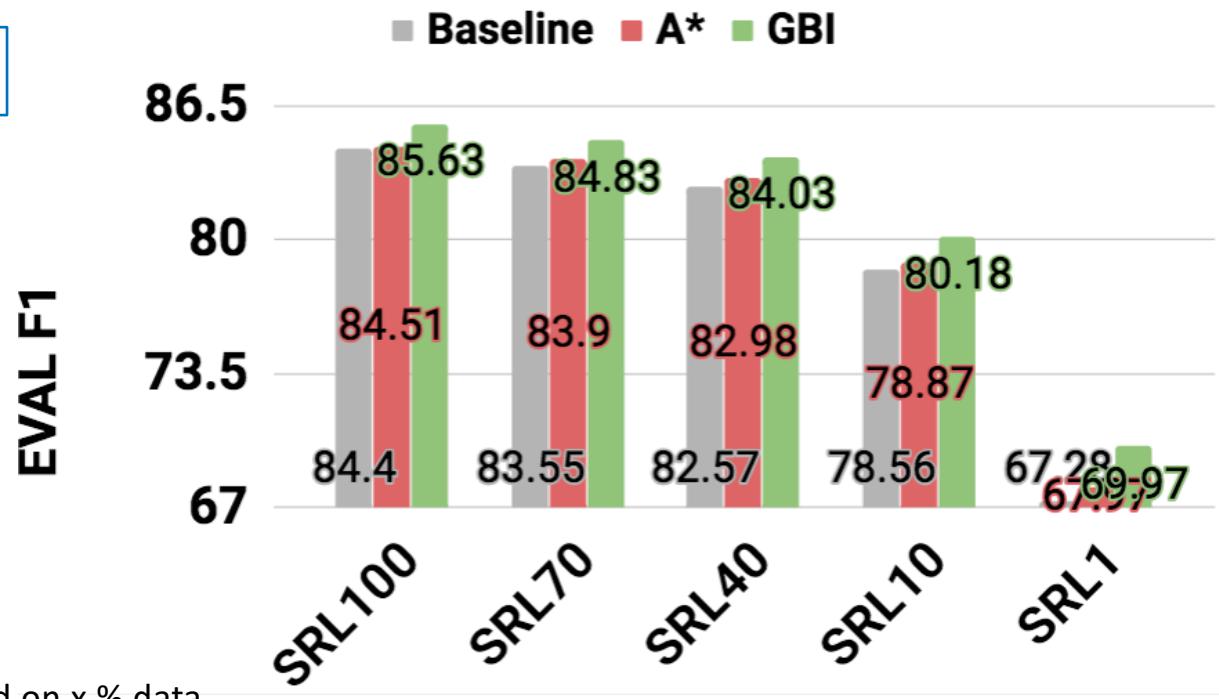
SRL x : SRL system trained on x % data.



GBI on SRL: Experiments

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

On Total test set

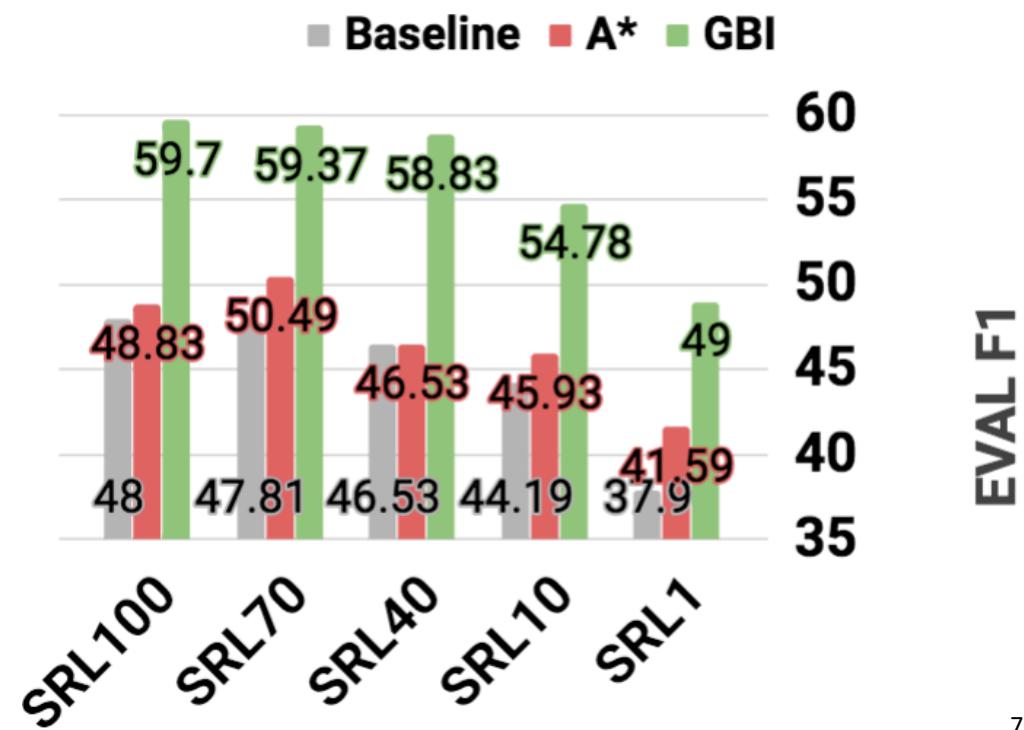
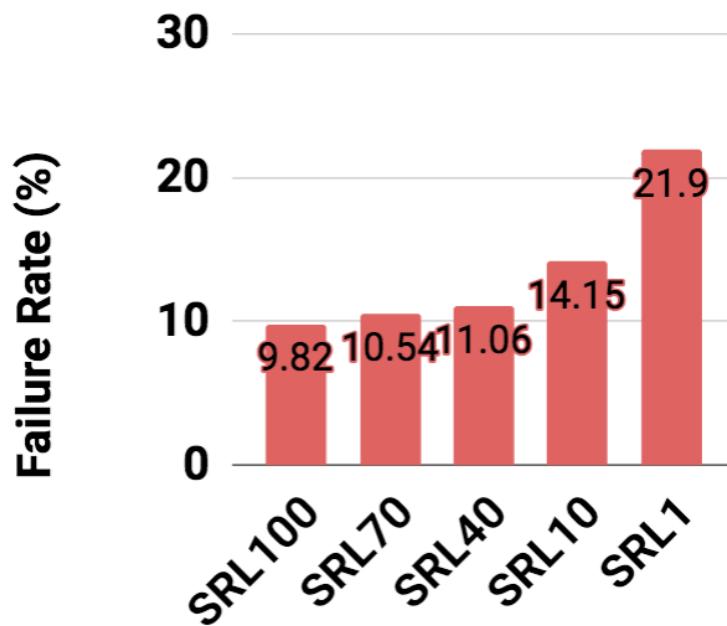


SRL x : SRL system trained on x % data.

GBI on SRL: Experiments

- Q4: Not sensitive to the baseline model performance

On Failure Set

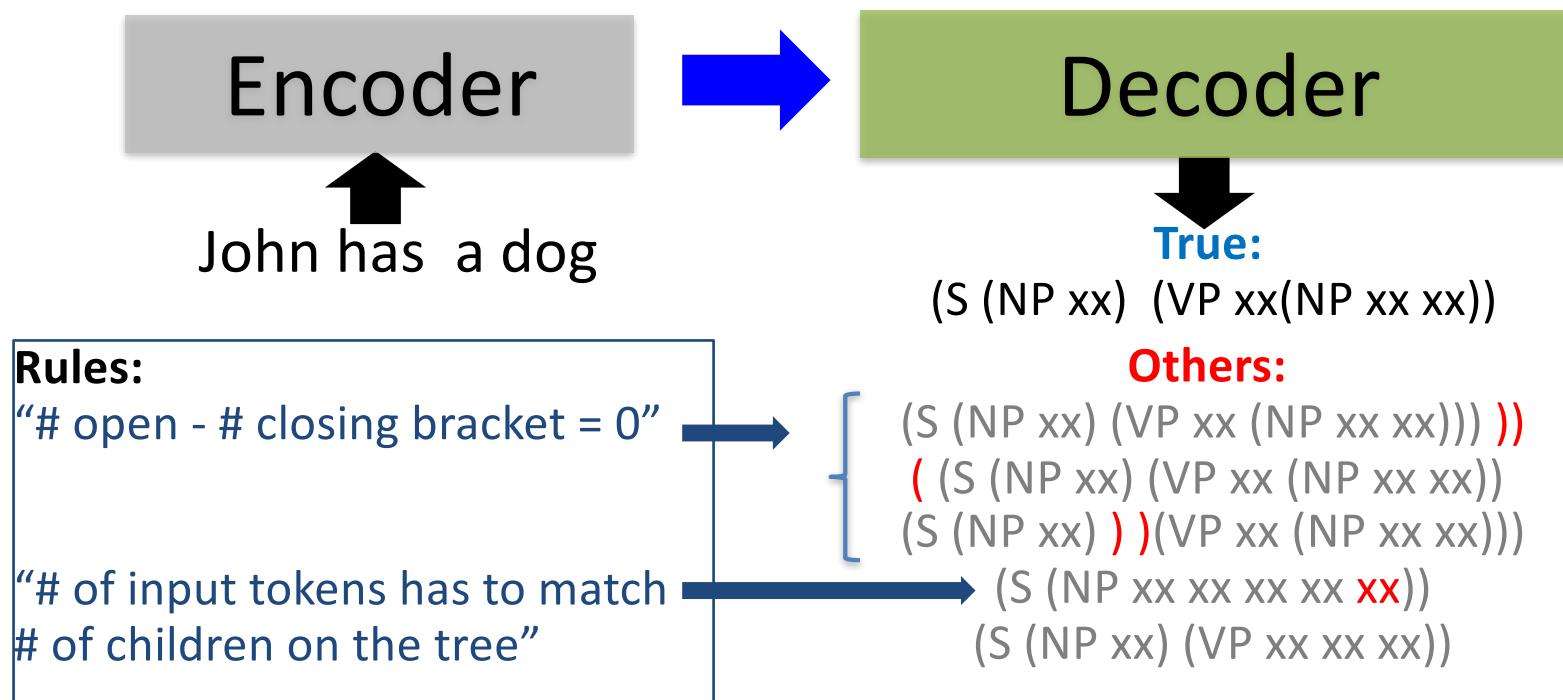


Application 2: Syntactic Parsing

- **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:

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

GBI on parsing: Experiments

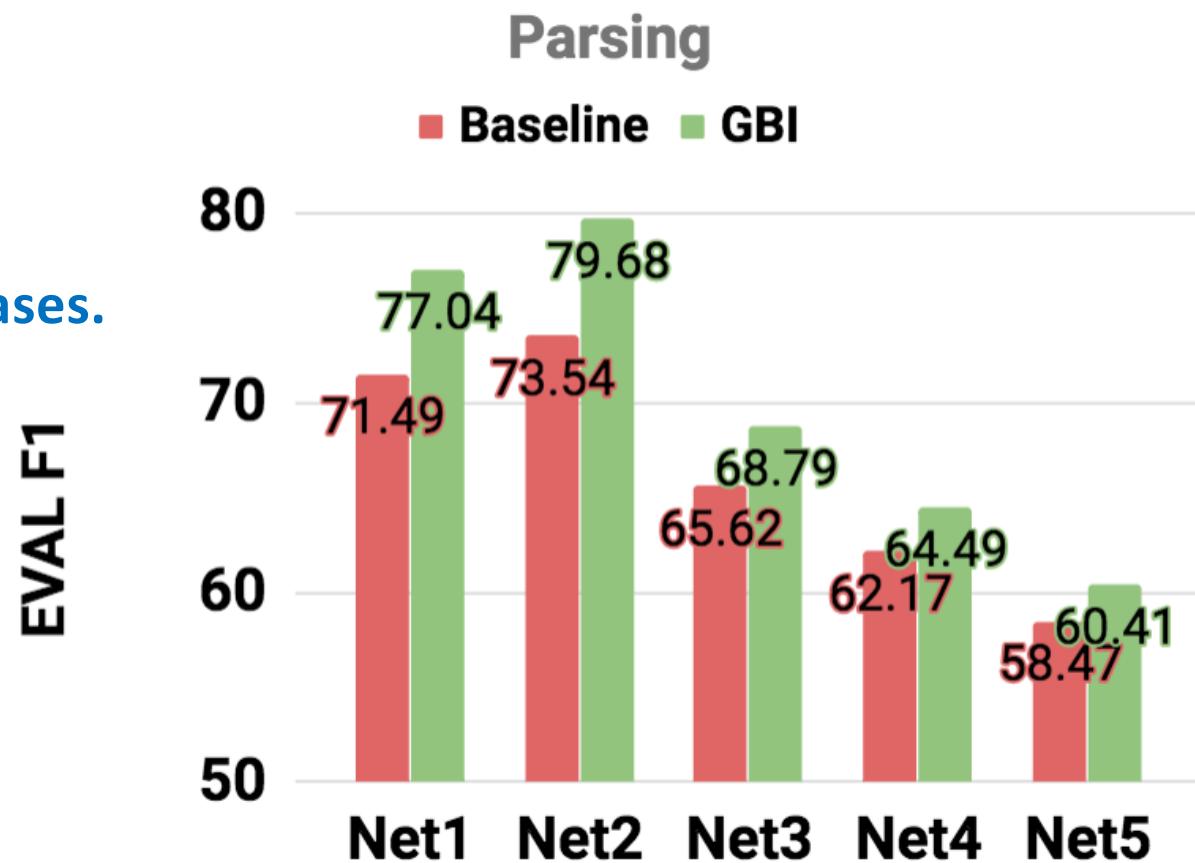
- Trained 5 parsers to answer Q1-4 (especially Q3)
 - Net1-2: GNMT (bi-directional encoder, uni-directional decoder)
 - Net3-5: uni-directional encoder, decoder.
 - Different dropout, used different portion of the dataset.

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	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	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%

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

S: azazbzazbzazbzbzbzbz



T: aaaaaazbaaabzbzaazbzbzbz

$$\begin{aligned} T: az &\rightarrow \textcolor{blue}{aaa} \\ bz &\rightarrow \textcolor{blue}{zb} \end{aligned}$$

constraint:
 $3 \times (\# a \text{ in } S) = \# a \text{ in } T$

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.*

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

S: azazbzazbzazbzbzbzbz

T: az → aaa
bz → zb

constraint:
 $3 \times (\# \text{ a in } S) = \# \text{ a in } T$

T: aaaaaazbaaazbzbaaazbzbzbzbz

GBI: extra considerations

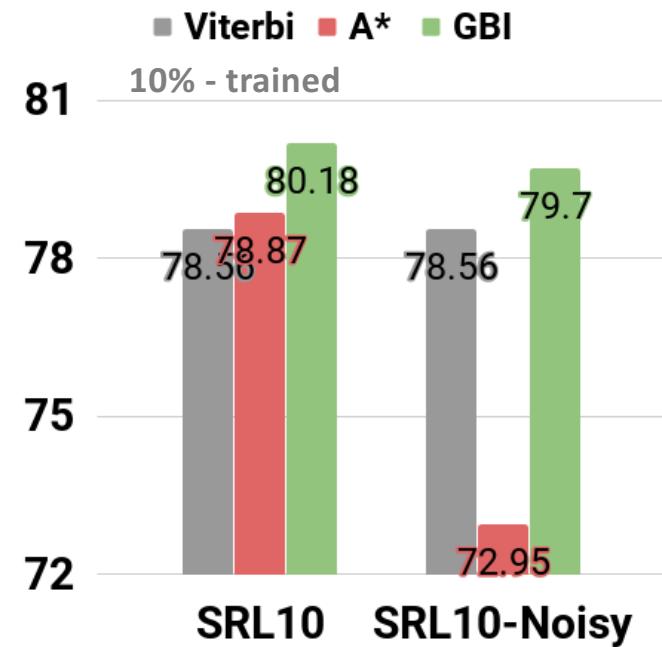
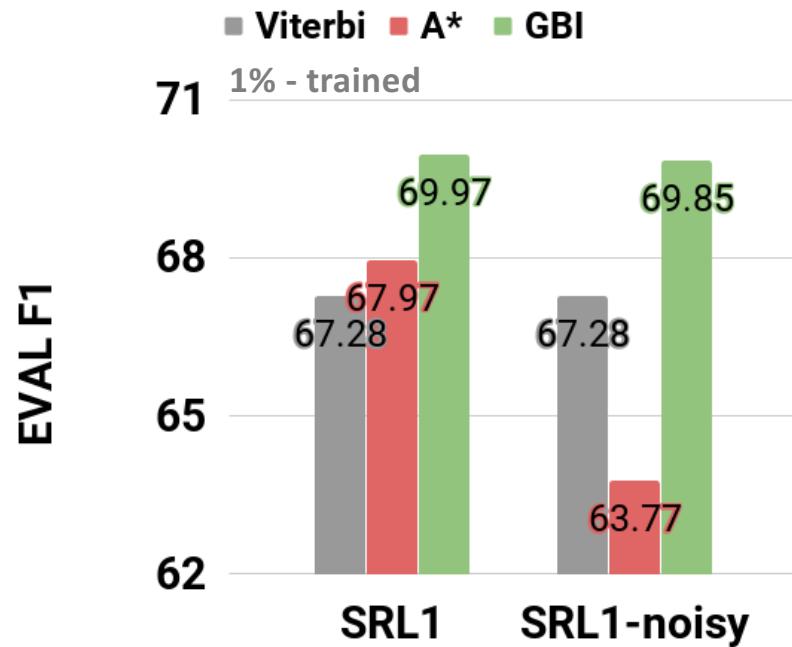
- Robustness to noisy constraint.
- Q5) out-of-domain performance.

Robustness of GBI

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

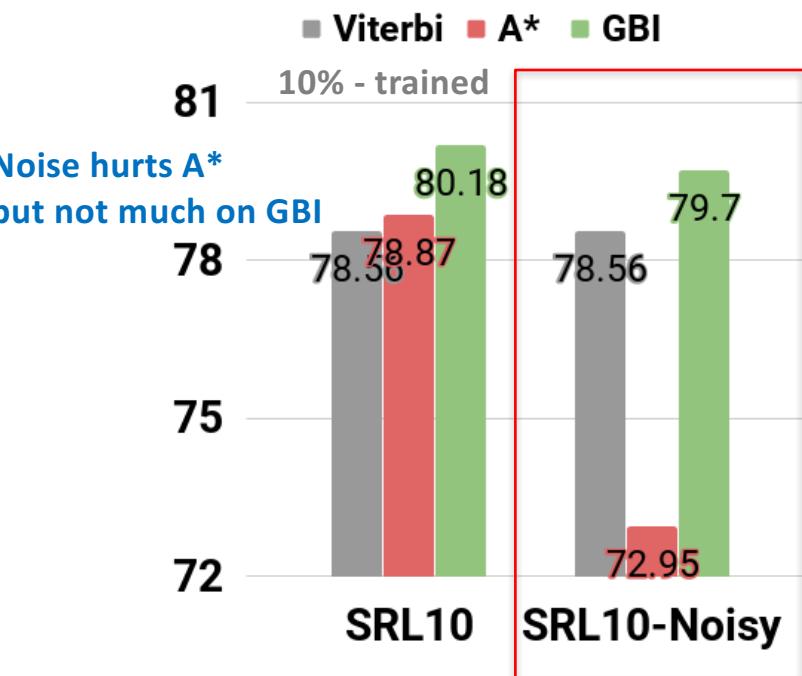
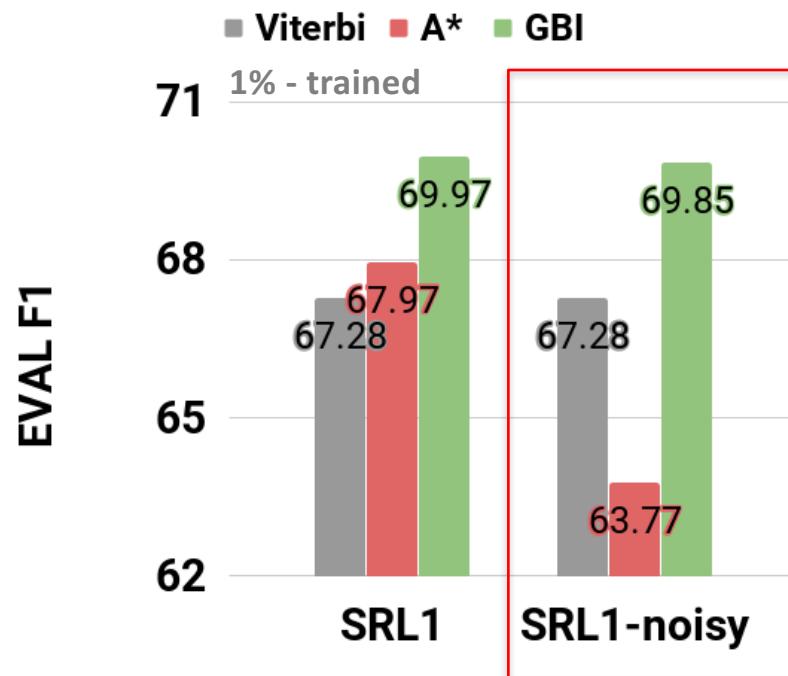
Robustness of GBI

- 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)

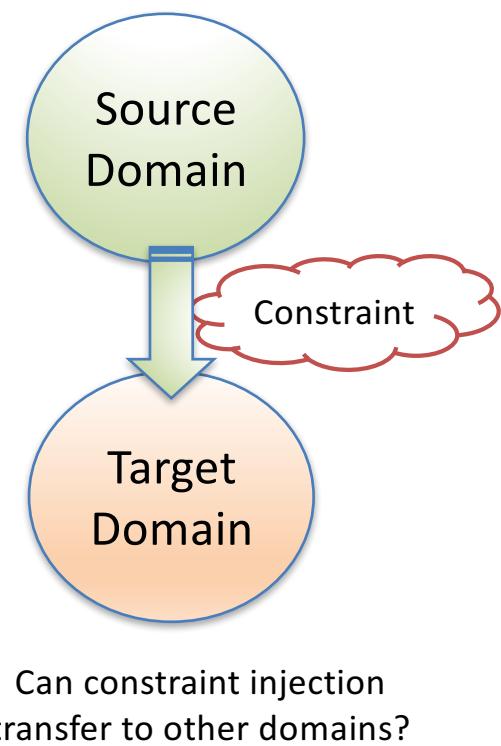


GBI on out-of-domain data (Q5)

- 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)



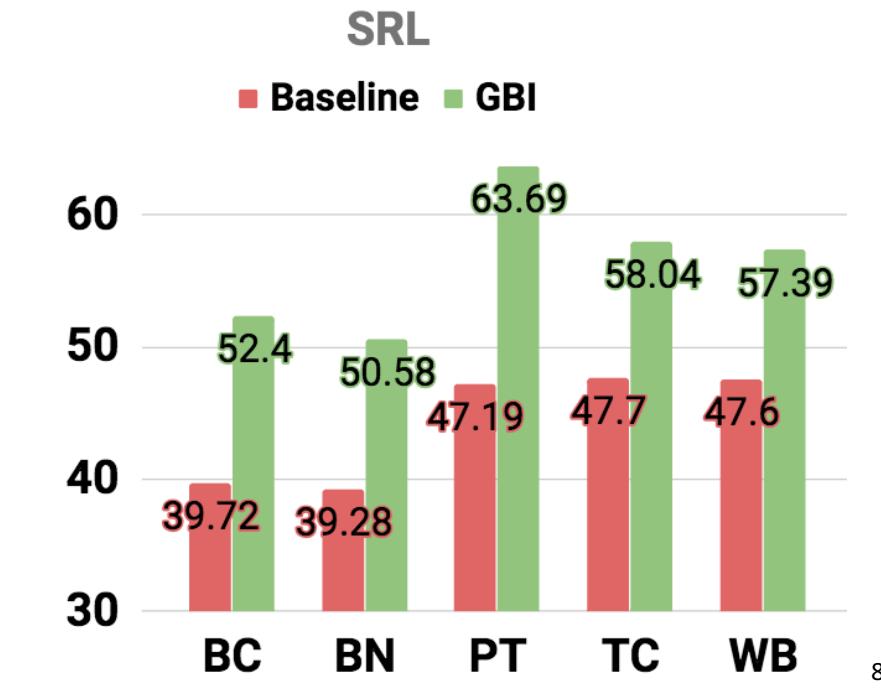
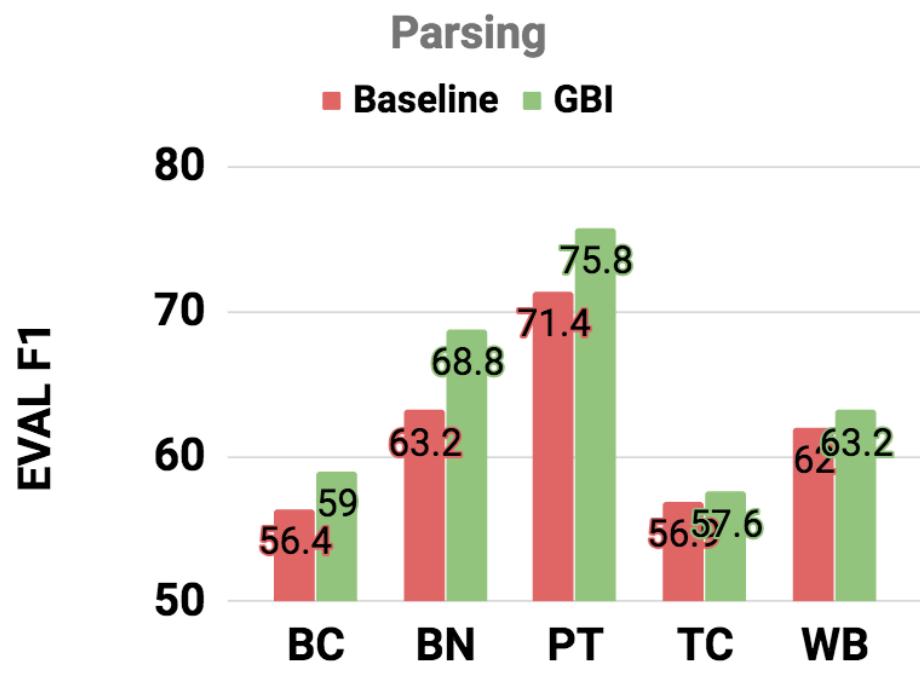
GBI on out-of-domain data

- 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)
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.



Research Questions

- Q1. How bad is the **baseline** model in terms of **constraint satisfaction**?
10% failure, 16% failure. (In-domain), ~15%, ~20%
- Q2. Does GBI actually *enforce constraints*?
 - ✓ • Yes!
- Q3. Does GBI *enforce constraints*
without compromising the quality of the system?
 - ✓ • Yes. It rather Improves!
- Q4. Is GBI *sensitive* to the **baseline model**?
 - ✓ • Not on 3 applications (5 experiment settings)
- Q5. *Performance* of GBI on *out-of-domain* dataset?
 - ✓ • Yes!

Roadmap

- Applications & constraints

- Inference with constraints [AAAI 19']

- **Learning with constraints [EMNLP 18']**

- Related Work

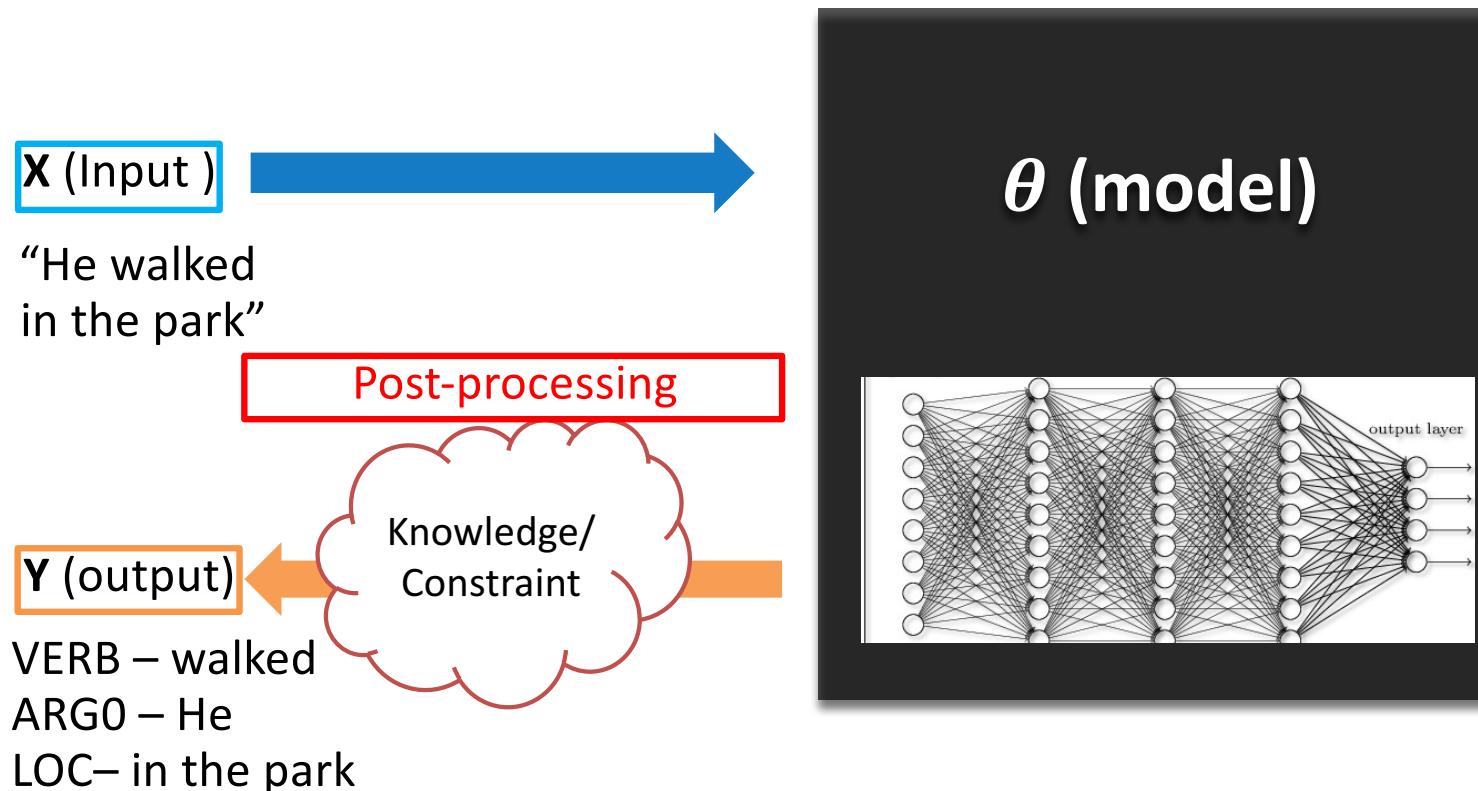
- Proposed Work

- New methods (exploration).
 - New application.

Completed
Work

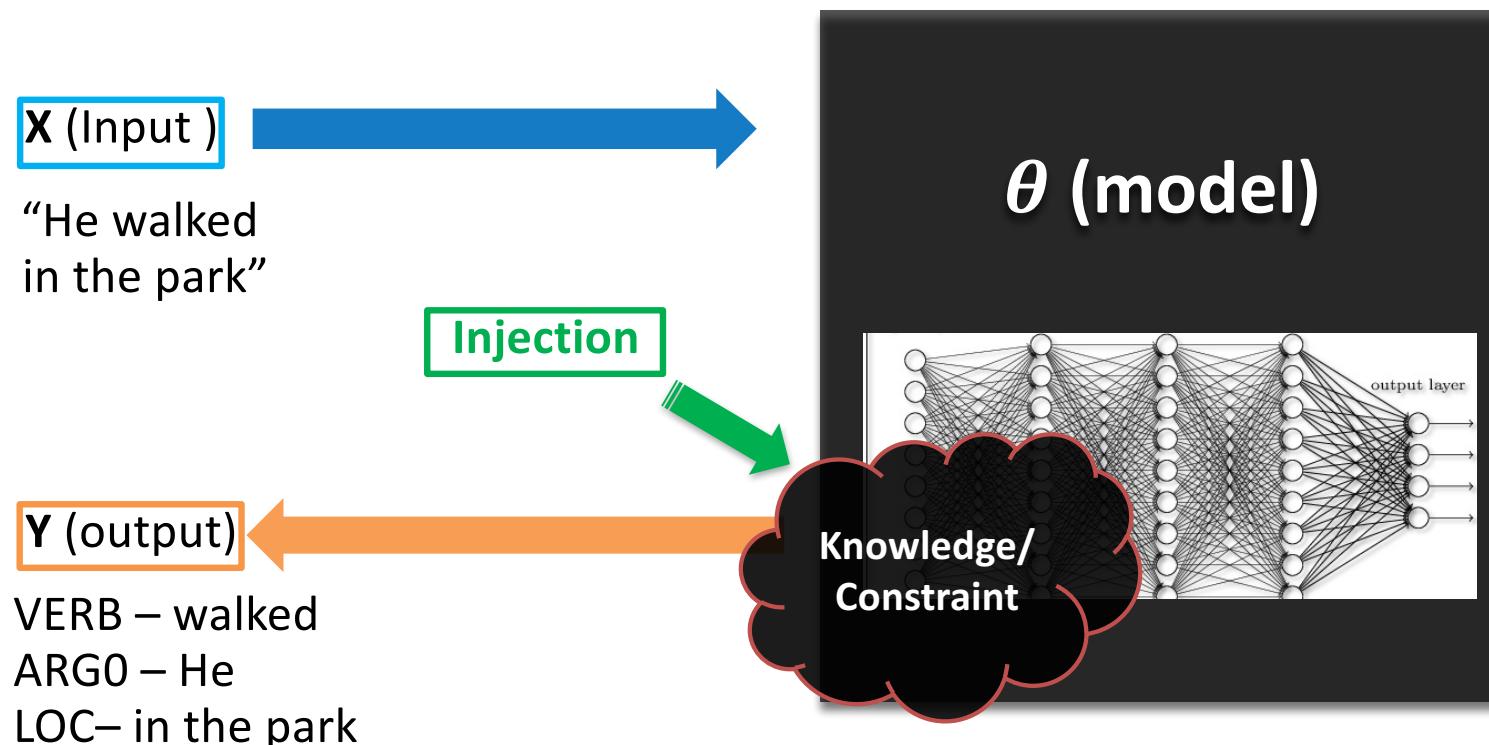
Learning with constraints

- Standard approach: “*Post-processing*”



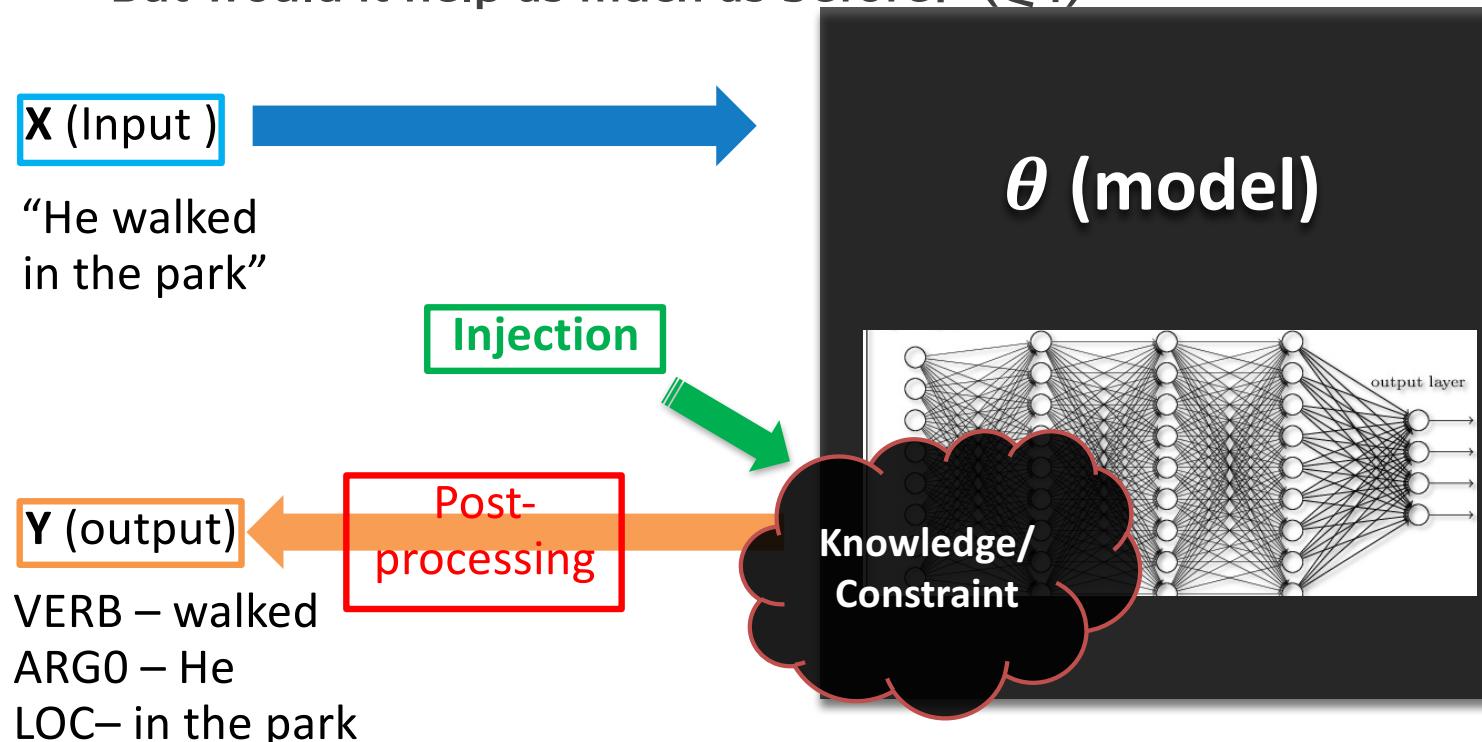
Learning with constraints

- “Injection of knowledge” beyond “Post-processing”.



Learning with constraints

- Post-processing still can be done the same way.
- But would it help as much as before? (Q4)



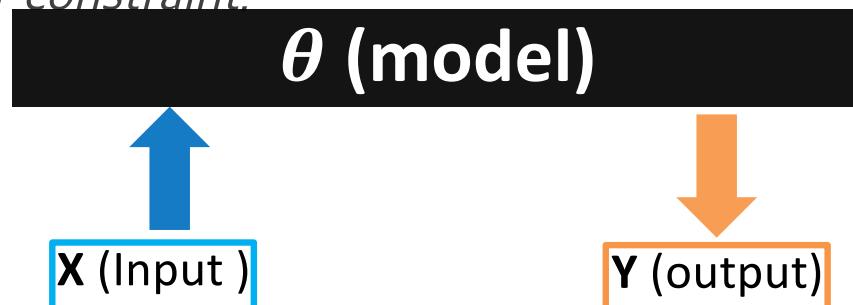
Learning with constraints

- Standard Learning
 - minimizing negative log-likelihood

$$-\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}} \sum_{i=1}^{|\mathbf{y}|} \log p(y_i | \mathbf{x}; \mathbf{w})$$

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

Motivation:
Can we use how well
(or how bad) the model
satisfies the constraint as a
signal to learn further?



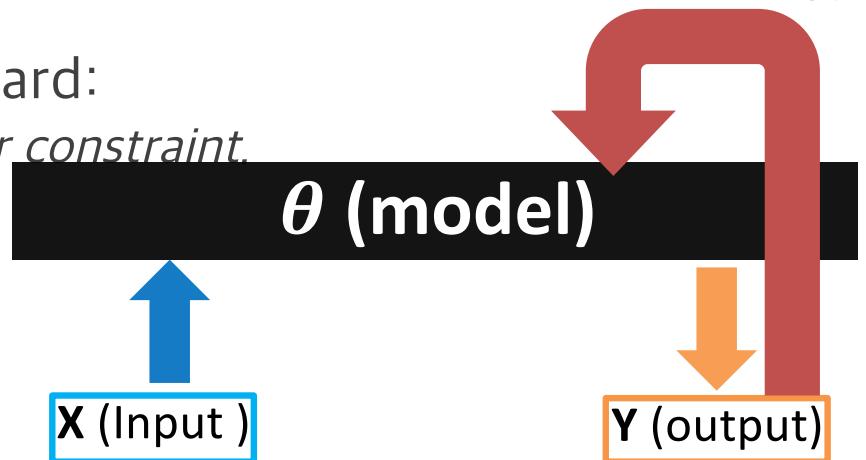
Learning with constraints

- Standard Learning
 - minimizing negative log-likelihood

$$-\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}} \sum_{i=1}^{|\mathbf{y}|} \log p(y_i | \mathbf{x}; \mathbf{w})$$

*reward * $\nabla \log(P(y))$*

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



Learning with constraints

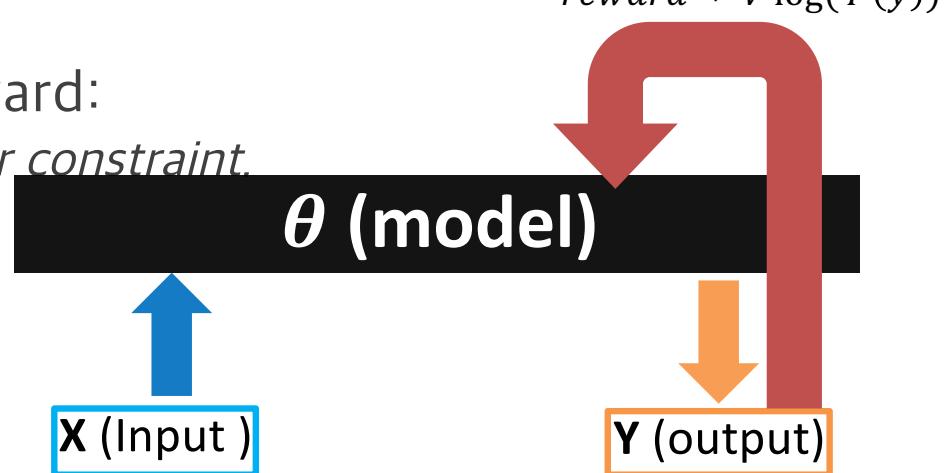
- Standard Learning
 - minimizing negative log-likelihood

$$-\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}} \sum_{i=1}^{|\mathbf{y}|} \log p(y_i | \mathbf{x}; \mathbf{w})$$

Challenge:

Which instances do we examine?
If we sample, we are likely to
constraint-violating sets.

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



Learning with constraints

- Standard Learning
 - minimizing negative log-likelihood

$$-\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}} \sum_{i=1}^{|\mathbf{y}|} \log p(y_i | \mathbf{x}; \mathbf{w})$$

- Learning with the reward:
 - Reward $s(x, y)$ defined by our constraint.

$$s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}; \mathbf{w}^{(t)})$$

Learning with constraints

- Learning with the reward $s(x,y)$
 - Inconsistency loss

$$s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}; \mathbf{w}^{(t)})$$

- Joint loss
- Semi-supervised
Joint loss

Learning with constraints

- Learning with the reward $s(\mathbf{x}, \mathbf{y})$
 - constraint loss

$$s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}; \mathbf{w}^{(t)})$$

- Joint loss

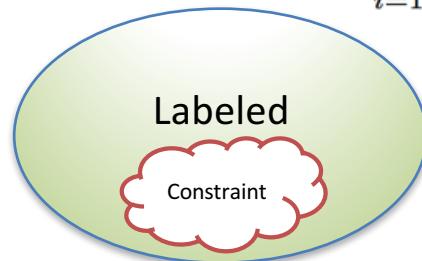
$$\overbrace{-\alpha_1 \sum_{i=1}^{|y|} \log p(y_i | \mathbf{x}_k; \mathbf{w}) + \alpha_2 s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}_m; \mathbf{w}^{(t)})}^{\text{SRL-}\underline{\text{labeled}}\text{ data } (\mathbf{x}_m = \mathbf{x}_k)}$$

Learning with constraints

- Learning with the reward $s(\mathbf{x}, \mathbf{y})$
 - constraint loss (c-loss)

$$s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}; \mathbf{w}^{(t)})$$

- Joint loss

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


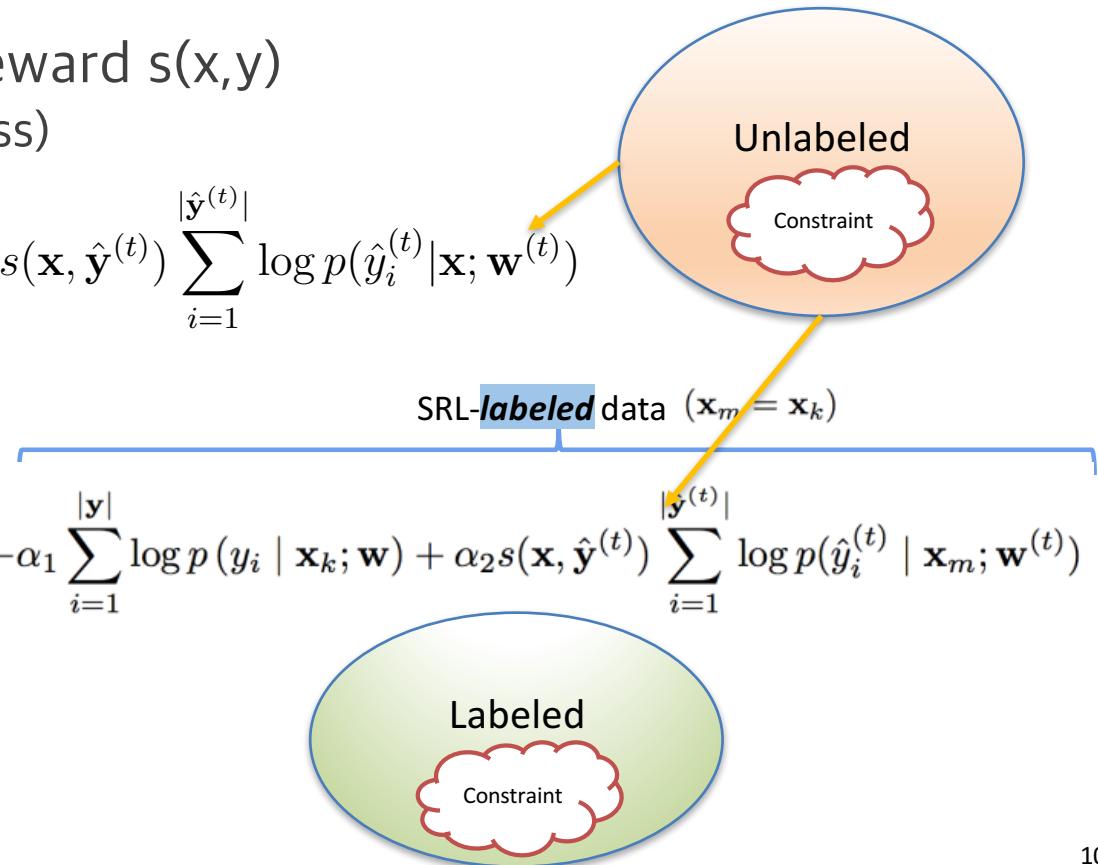
Learning with constraints

- Learning with the reward $s(\mathbf{x}, \mathbf{y})$
 - constraint loss (c-loss)

$$s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}; \mathbf{w}^{(t)})$$

- Joint loss

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

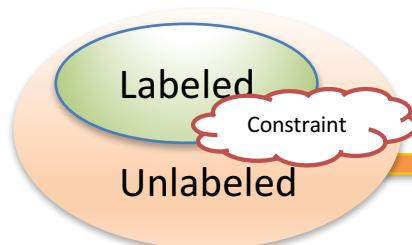


Learning with constraints

- Learning with the reward $s(\mathbf{x}, \mathbf{y})$
 - constraint loss (c-loss)

$$s(\mathbf{x}, \hat{\mathbf{y}}^{(t)}) \sum_{i=1}^{|\hat{\mathbf{y}}^{(t)}|} \log p(\hat{y}_i^{(t)} | \mathbf{x}; \mathbf{w}^{(t)})$$

- Joint loss
- Semi-supervised Joint loss



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

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

Learning with constraints

- Learning with the reward $s(\mathbf{x}, \mathbf{y})$
 - constraint loss (c-loss)

Unlabeled
Constraint

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

- Joint loss

Labeled
Constraint

Semi-supervised
Joint loss

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

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

SRL-***labeled*** data

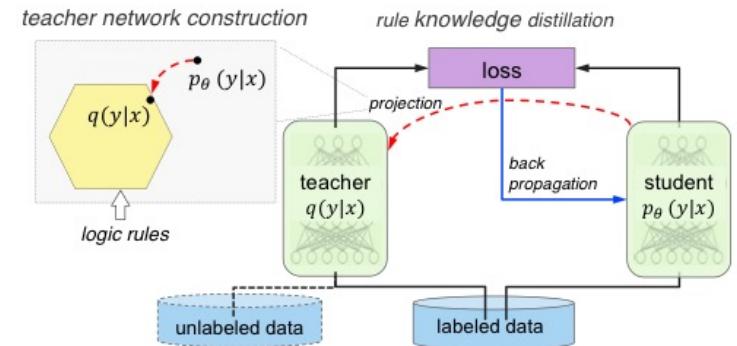
SRL-***unlabeled*** data ($\mathbf{x}_m \neq \mathbf{x}_k$)

Labeled
Constraint
Unlabeled

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

Related Work

- Posterior regularization (PR) has a similar form as an joint objective.
- The philosophy is different as I directly give reward to outputs whereas PR will make a soft logit vector from the constraints.
- One drawback of this is that the base distribution $P(\cdot)$ is only stable for problems that current output does not depend on previous output (no seq2seq). (Hu 2016)



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

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

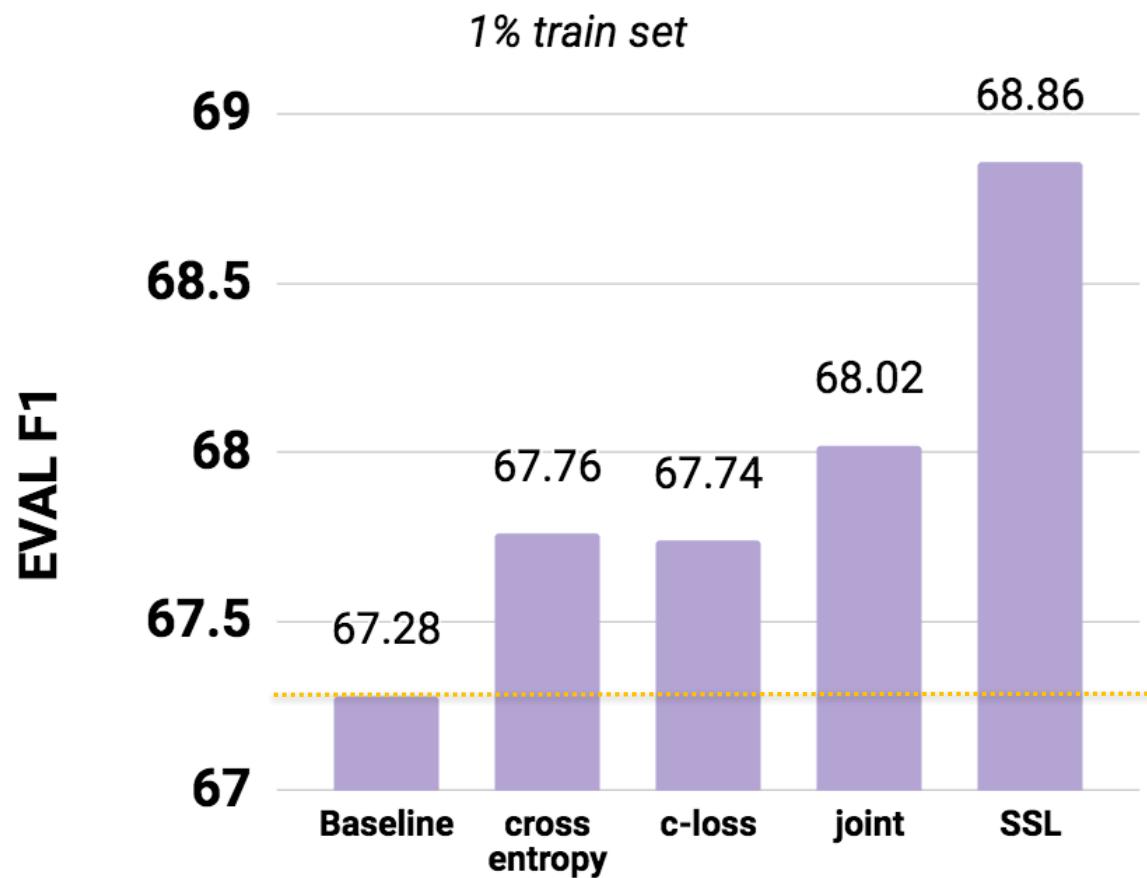
Research Questions

- Q1. Does *proposed method* actually *enforce* constraints?
- Q2. Does proposed method *enforce* constraints *without compromising* the quality of the system?
- Q3. How well does *SSL formulation perform*, especially in *low-resource* scenarios?
- Q4. Is the train time & decoding time effort mutually exclusive?
(training time vs. decoding time?)

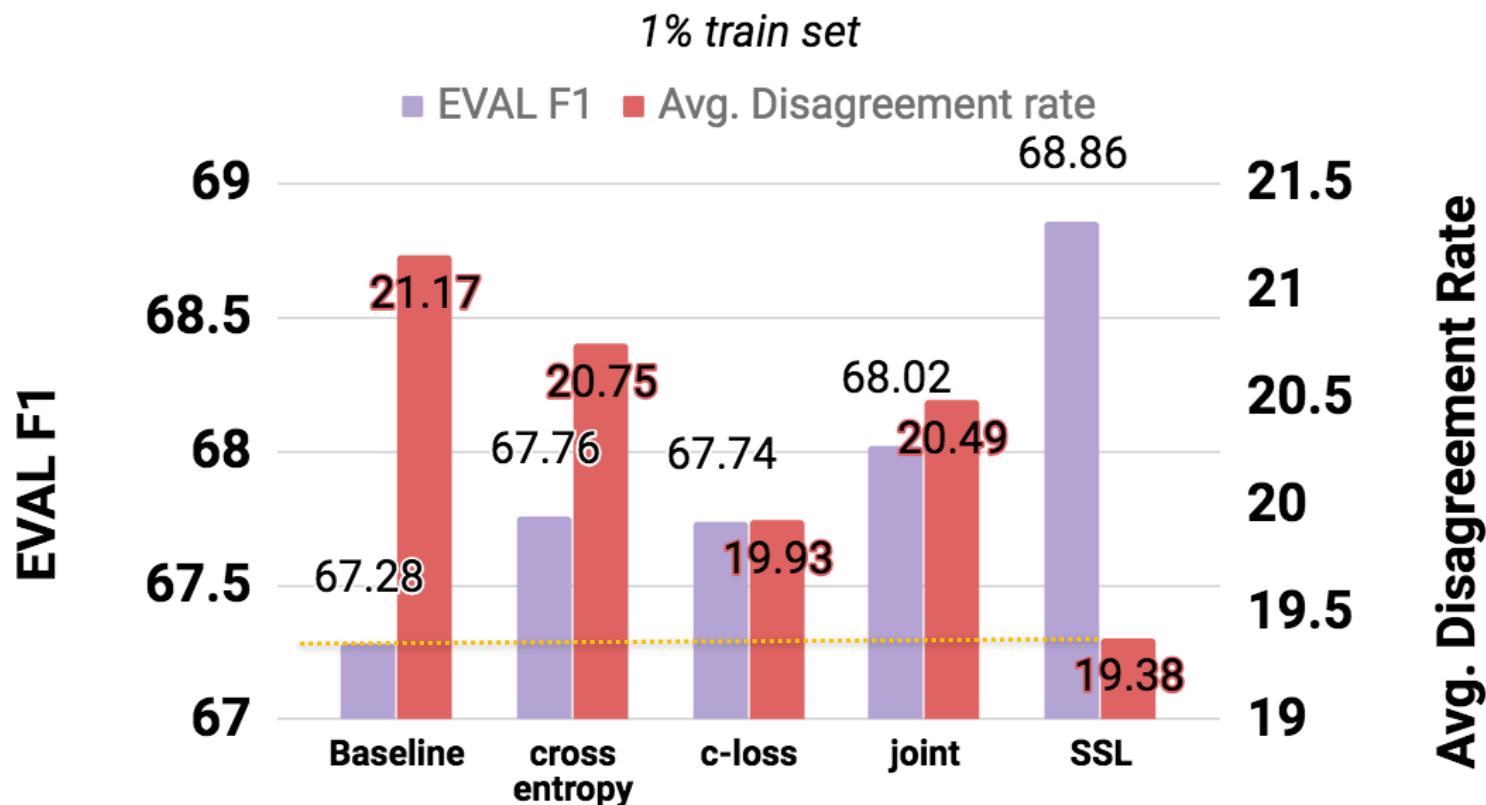
SRL Experiments (again)

- 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)

Comparison of losses (on 1%)

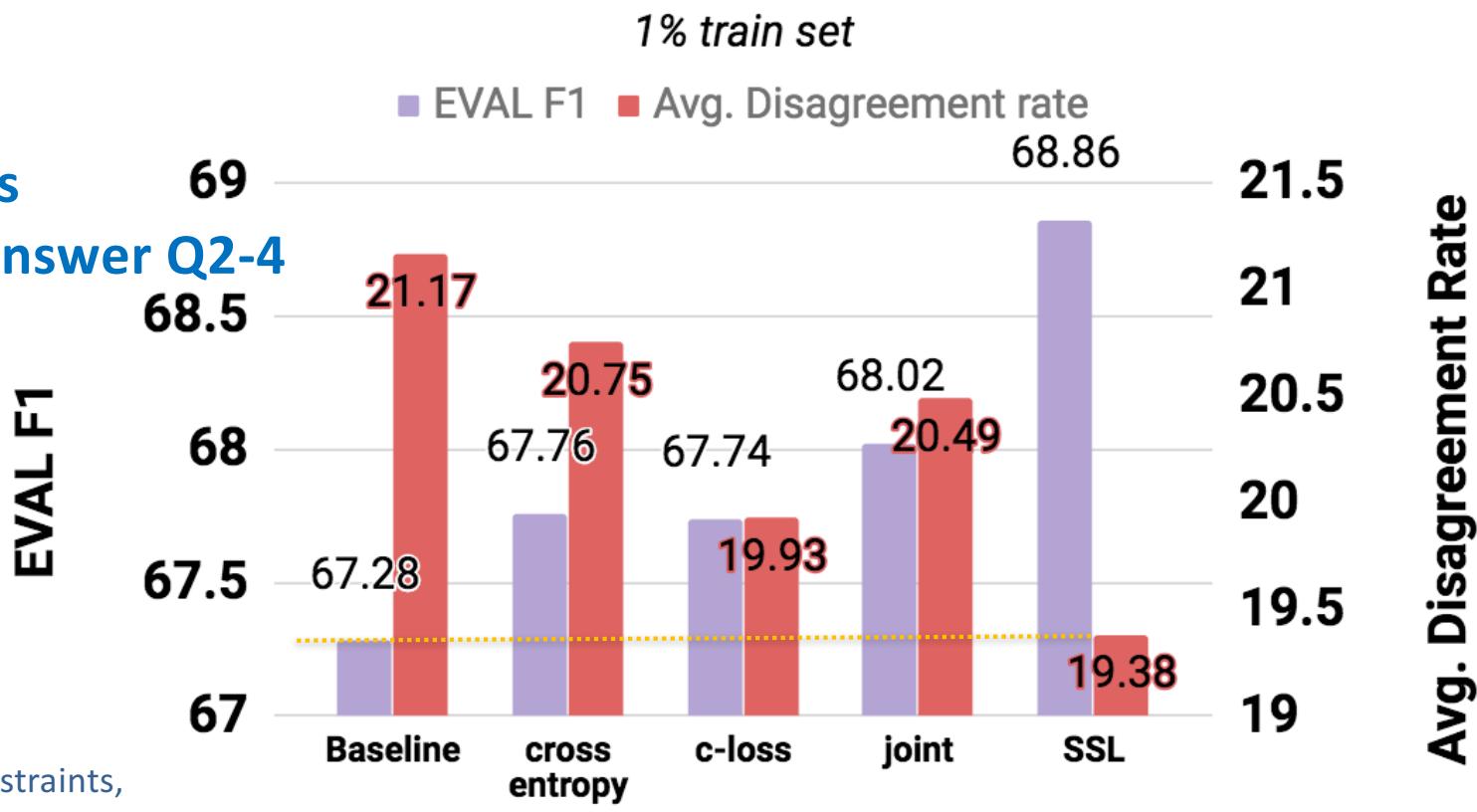


Comparison of losses (on 1%)



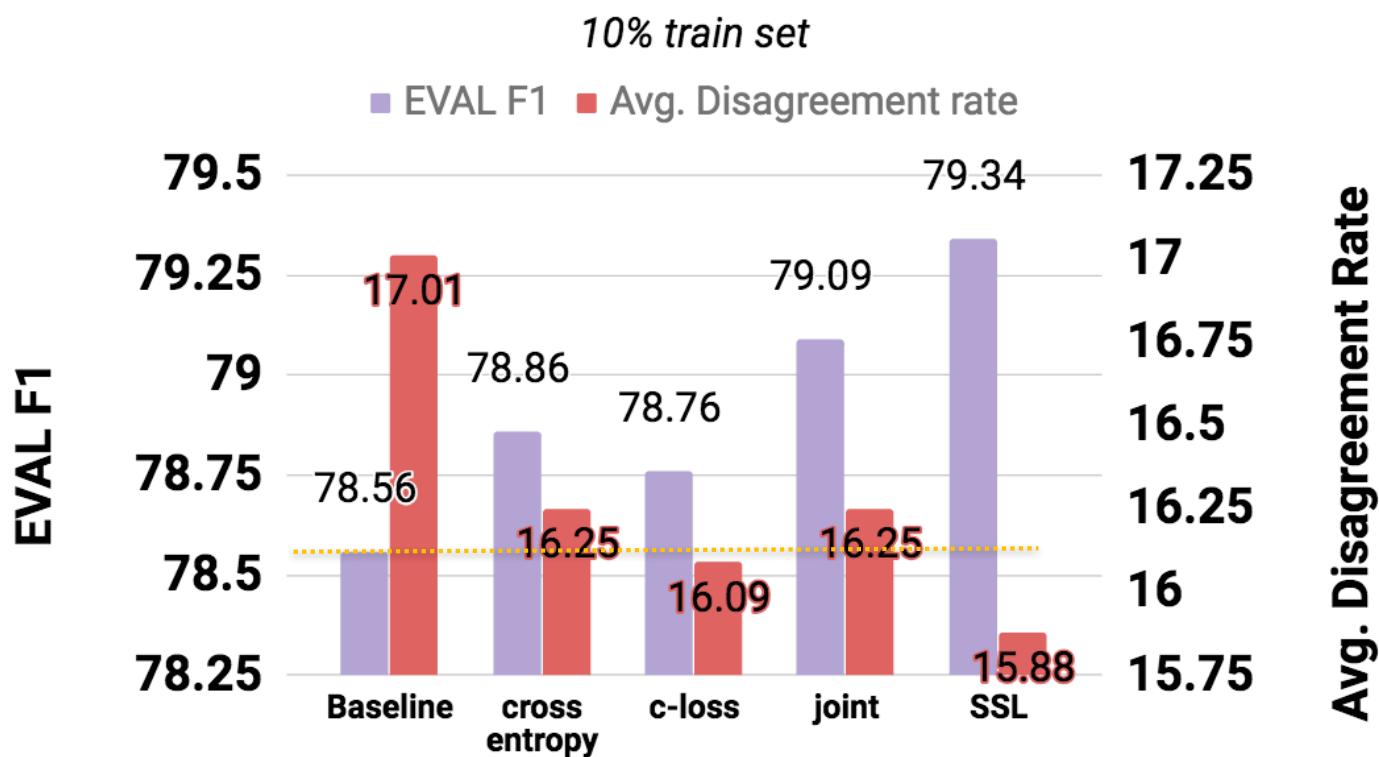
Comparison of losses (on 1%)

All the loss
function answer Q2-4
favorably!



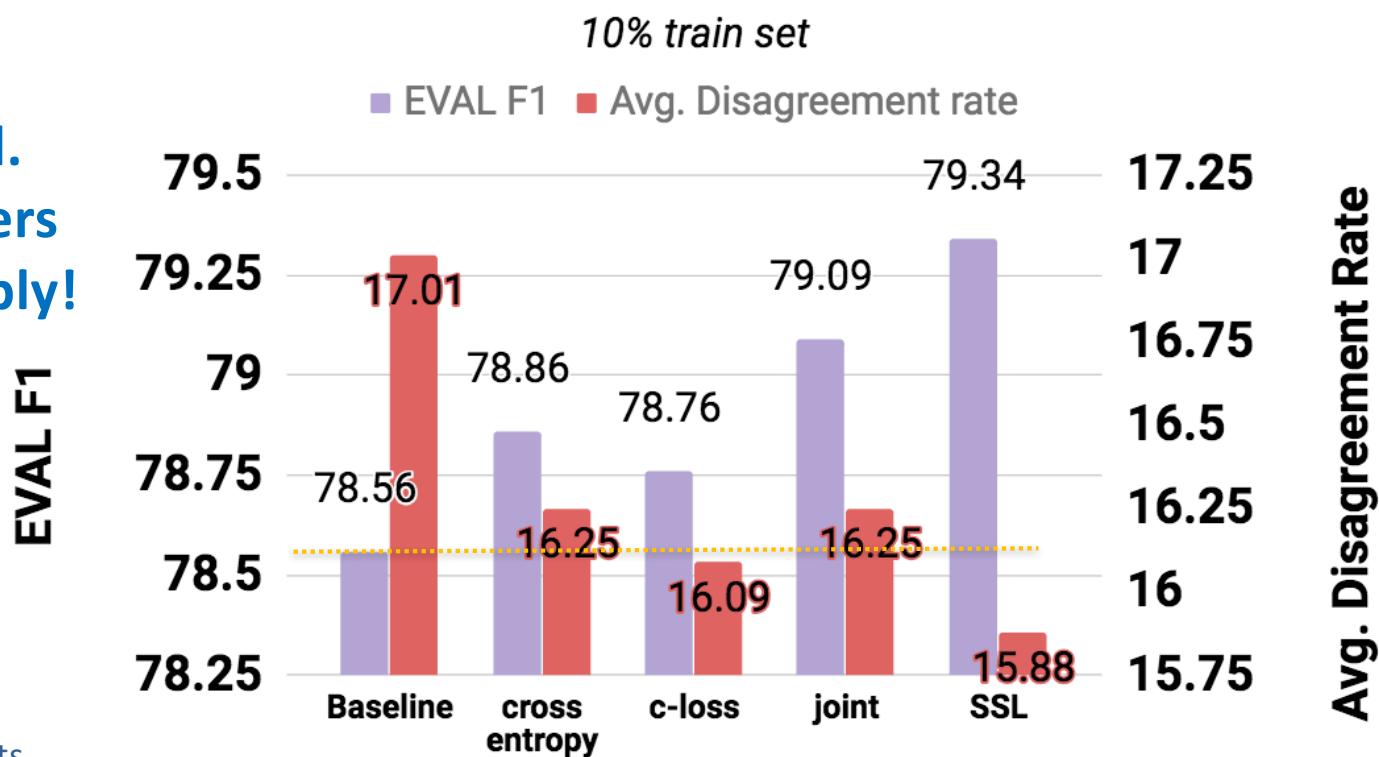
Q2. enforce constraints,
Q3. enforce constraints *without compromising* the quality,
Q4. SSL formulation perform well?

Comparison of losses (on 10%)



Comparison of losses (on 10%)

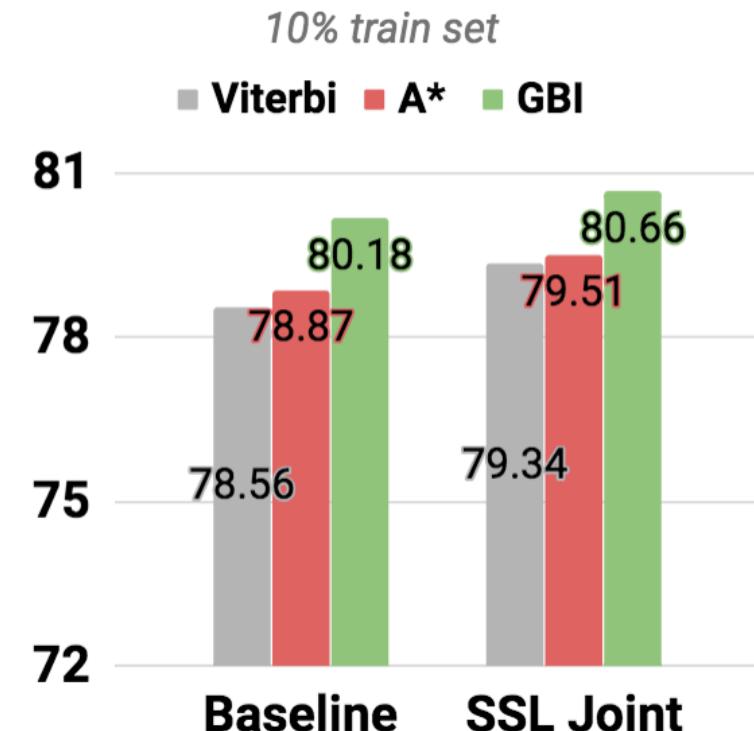
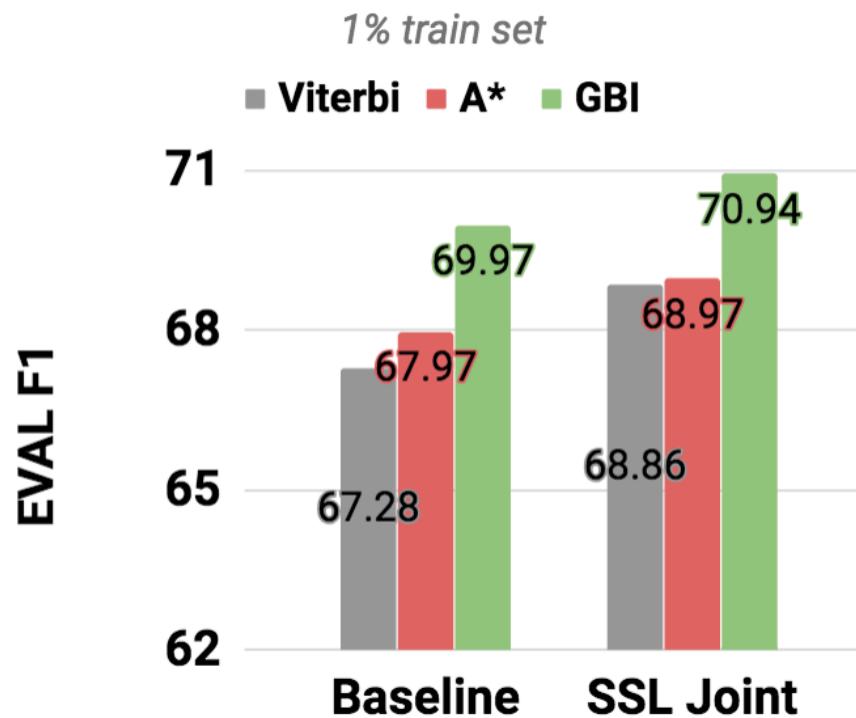
Similar trend.
Again, answers
Q2-4 Favorably!



Q2. enforce constraints,
Q3. enforce constraints **without compromising** the quality,
Q4. SSL formulation perform well?

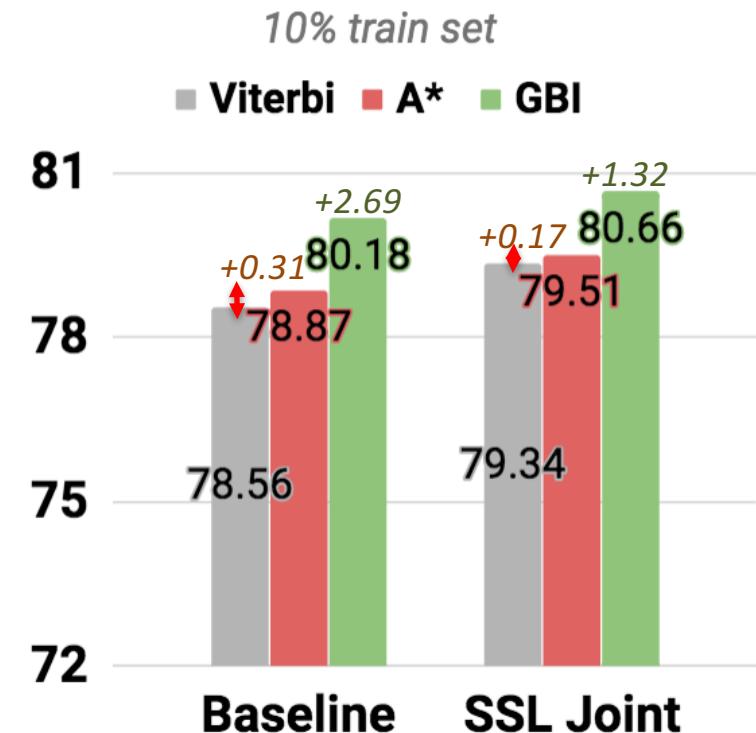
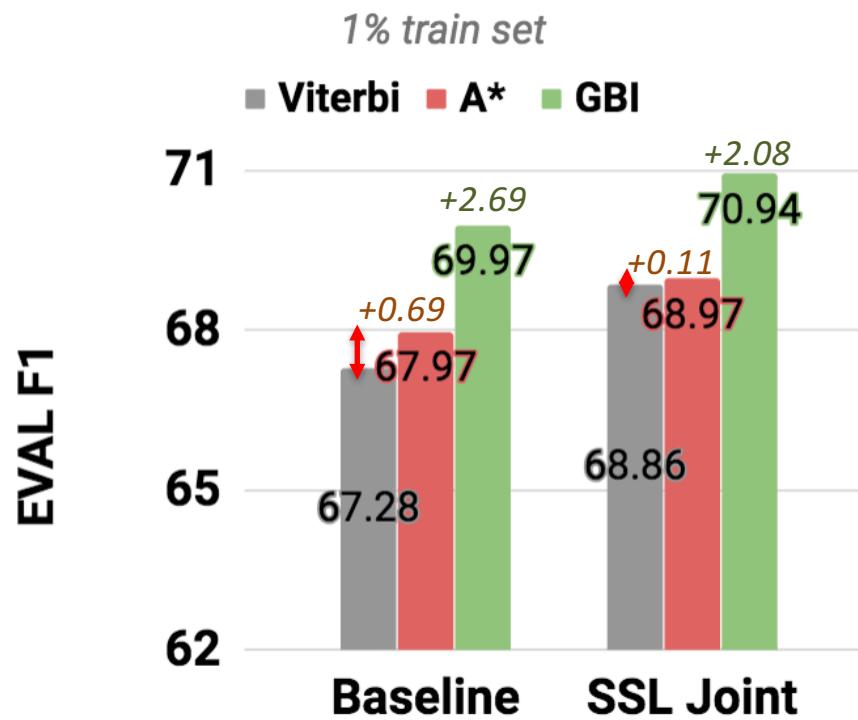
Q5 train & test are complementary?

On 1% dataset: GBI brings additional gains on top of train-time constraint injection.



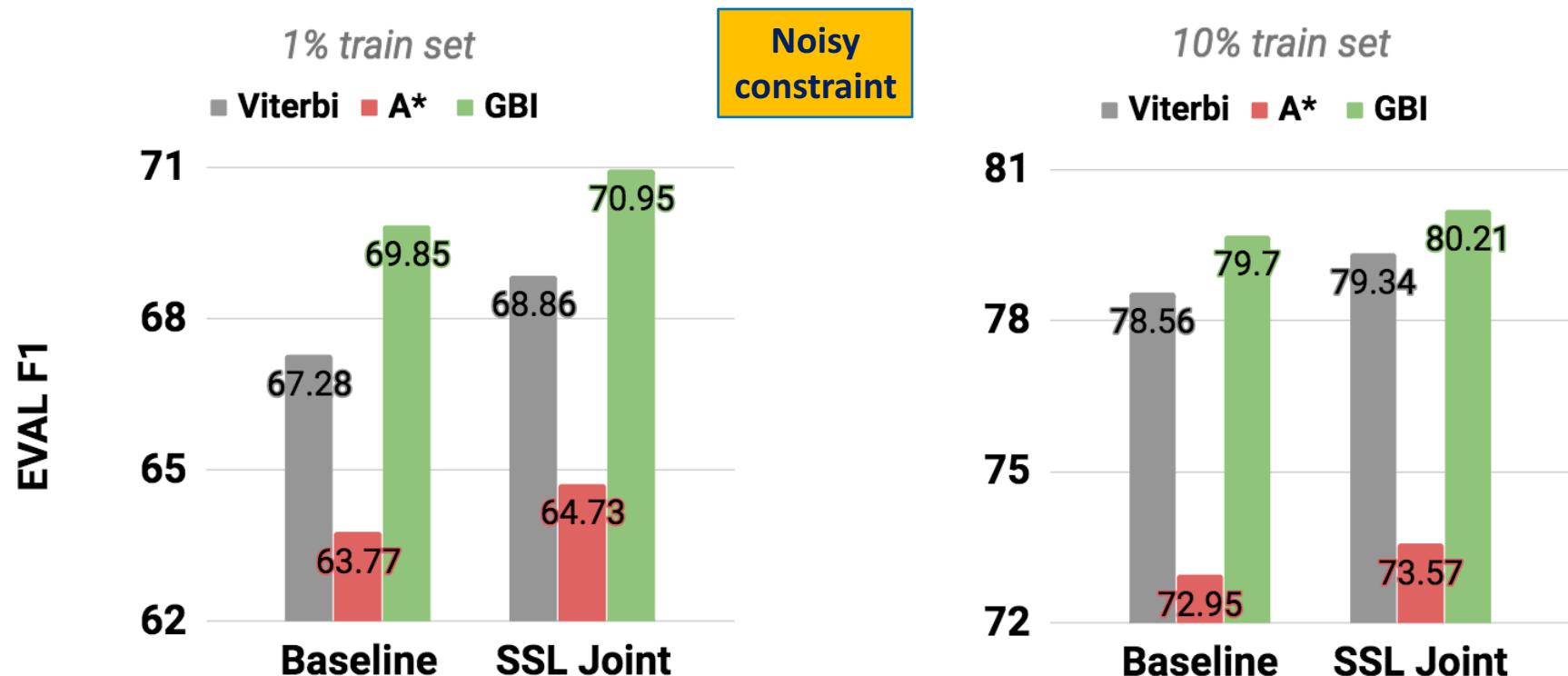
Q5 train & test are complementary?

On 1% dataset: GBI brings additional gains on top of train-time constraint injection.



Q5 Constraints on train vs. test.

Both on 1, 10%: GBI brings additional gains on top of train-time constraint injection



Summary: Learning with constraints

- Q1. Does *proposed method* actually *enforce* constraints?
✓ • Yes!
- Q2. Does proposed method *enforce* constraints
without compromising the quality of the system?
✓ • Yes, and improves as GBI did.
- Q3. How well does *SSL formulation perform*,
especially in *low-resource* scenarios?
✓ • It works the best out of all the losses proposed.
- Q4. Is the train time & decoding time effort mutually exclusive?
✓ (*training time vs. decoding time*?)
• No! The efforts add up. The two are complementary.

Summary on completed work

- Constraint Injection (ancillary objective)
 - Output constraint “satisfaction”
 - Main performance Improvement
 - GBI is faster & better than A* algorithm.
 - By leveraging “constraints” that do not require labels
 - Better performance on out of domain
 - Semi-Supervised Learning
 - [Proposed] Domain adaption (unlabeled target)

Conclusion

- General constraints
 - No differentiability requirement.
 - Simple (normalized) count of errors work.
 - Model-agnostic
 - Only changed “loss” function.
 - No separate feature engineering nor network structure change.
 - ‘Anyone’ can easily apply constraint.
 - Can quickly test out constraint using GBI.
 - The loss on GBI and Learn-time are similar
 - Examine potential effect of constraint injection with GBI before training another model.
- 
- Explored constraint function

 - Aggregate (sum) constraint function.
 - Transduction problem.
 - Constraint occurring from representation.
 - Brackets from parsing
 - Syntactic constraint from linguistic knowledge.
 - Semantic Role Labeling.

Roadmap

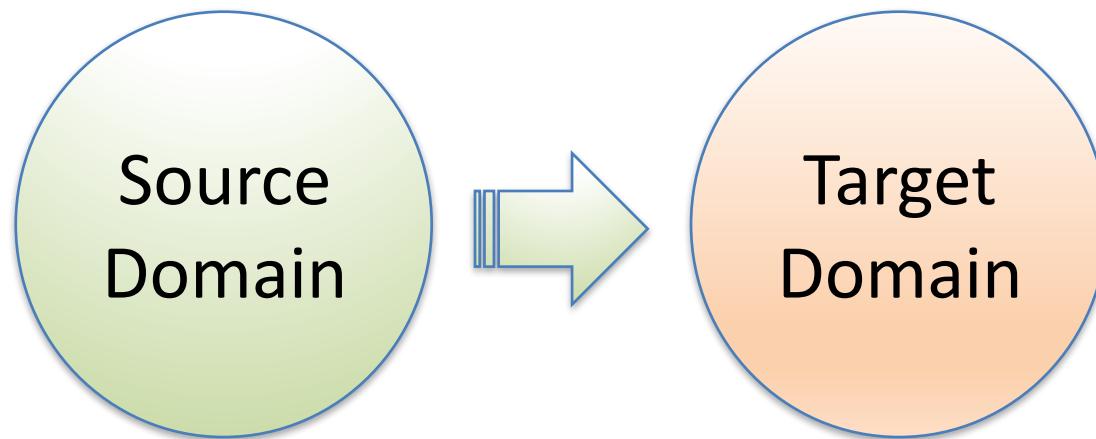
- Motivations & Background
- Inference with constraints
- Learning with constraints
- Proposed work
 - Application
 - new methods (exploration)

Completed Work

Proposed Work

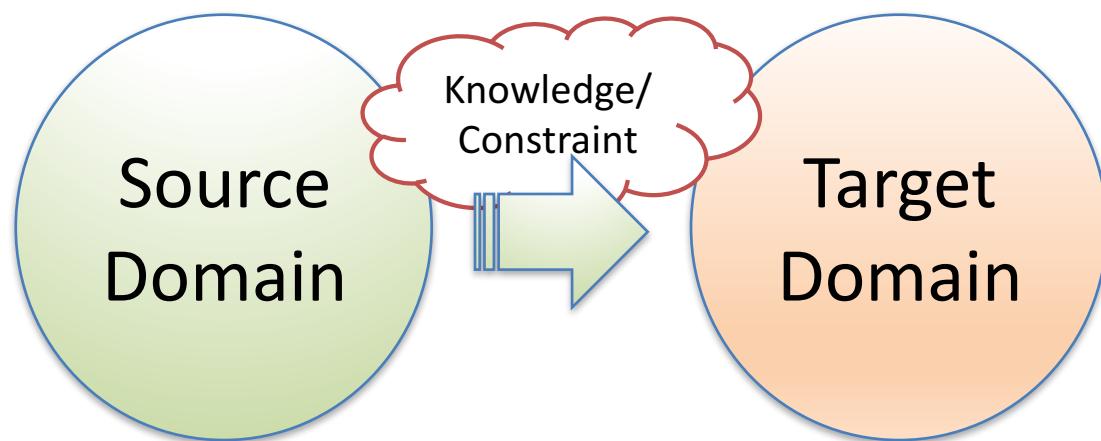
[App 1] Domain adaptation

- Leveraging unlabeled data
 - train on ‘source’ domain → target domain



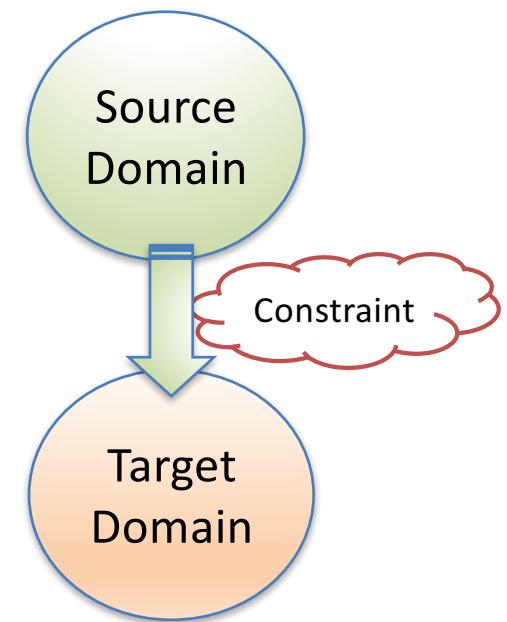
[App 1] Domain adaptation

- Leveraging unlabeled data
 - train on ‘source’ domain → target domain
 - Improve the target domain performance via constraint



CoNLL2012 domain again

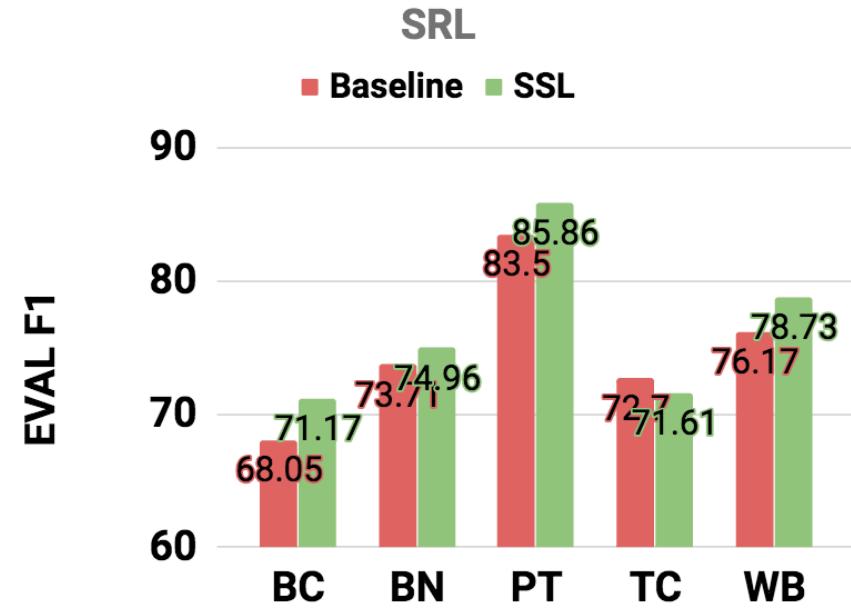
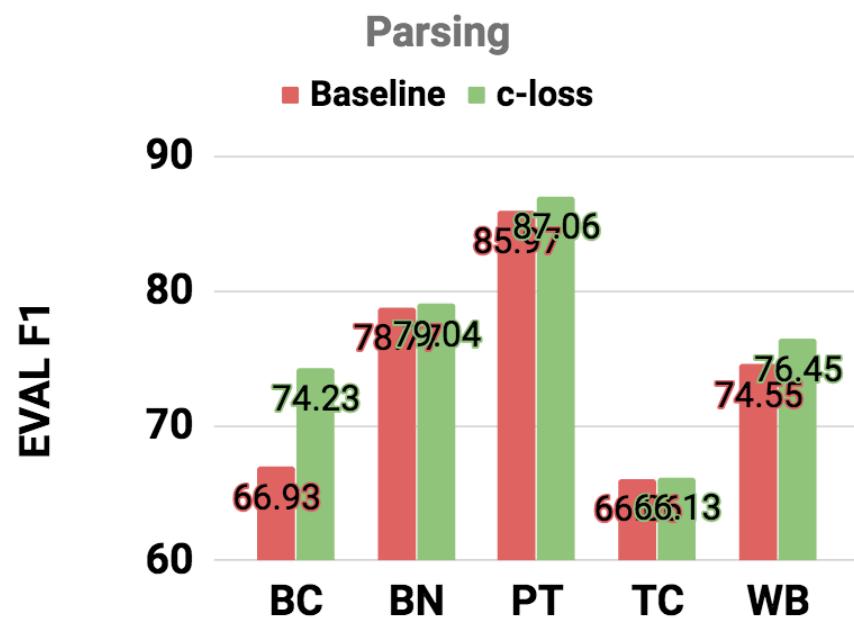
- CoNLL2012
 - NewsWire (NW) which includes WSJ
 - Broadcast Conversation (BC)
 - Broadcast News (BN)
 - Pivot Corpus (PT)
 - Telephone Conversation (TC)
 - Weblogs (WB)



Can constraint injection transfer to other domains?

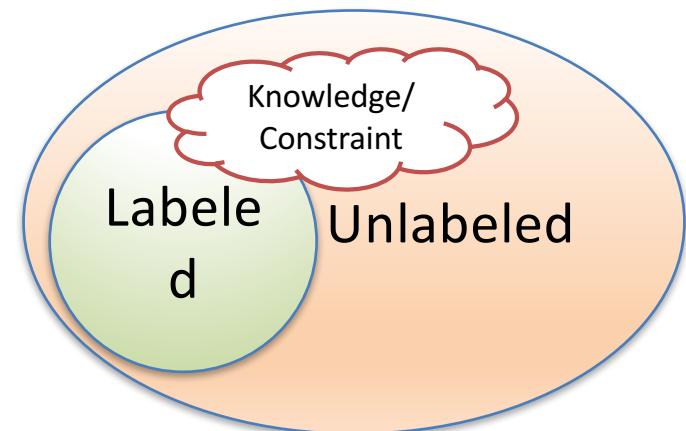
Preliminary Experiment

- Parsing with c-loss (constraint violation loss)
- SRL with SSL joint loss.



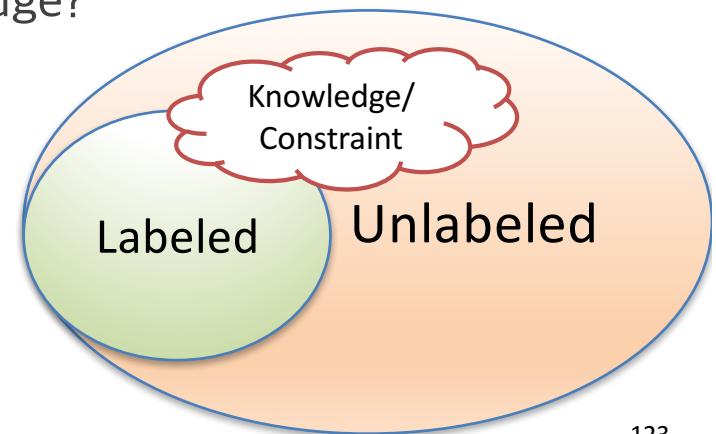
[App 2] “Any” span-based application

- There is no SRL-specific component to the earlier “learning with constraints”.
- Would this approach work on “general” span-based model?
 - Co-reference resolution
 - Framenet SRL
 - Shallow Discourse Parsing
 - ...



[App 2] “Any” span-based application

- Challenges to “Any” span-based models.
 - New models that consider all the possible spans.
- Questions to ask
 - How well would current approach work on new models?
 - Do we need other ways to inject knowledge?
- SOTA work
 - Still focuses on the gold labels only.



Roadmap

- Motivations & Background
- Inference with constraints
- Learning with constraints
- Proposed work
 - Application
 - new methods (exploration)

Completed Work

Proposed Work

[Method 1] Negative sampling

- Learning with constraint, we tried to find the best output given model.
- Evaluate ‘how well (bad)’ the output is in terms of the constraint violation. → Use this signal to further train.
- Can we do better?
- Can we systematically sample constraint violating sets and learn from it?
- **Challenges:**
 - Sampling a sequence (sentence) with a score distribution is not straightforward.
 - Proposed method: Sampling using MCMC method.
 - How to learn from the negative samples?
 - Proposed method: Max-margin training

Proposed Timeline

Topic	subtopic	2018			2019						
		Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
	SSL on span-based models										
Exploration	<i>Applications</i>										
	<i>Methods</i>	sampling multi-tasks with agreement									
Finalization	<i>Method & Applications</i>										
	Thesis writing										

Thank You!

Questions?