

## **Constrained Conditional Models in Natural Language Processing**

Seminar Report

By

**Anamay Tengse**

Under the guidance of

**Prof. Ganesh Ramakrishnan**

May 3, 2013

# Outline of the presentation

- Constrained Conditional Models
  - Introduction
  - Approaches for training and testing
- Constraints as prior knowledge
- Learning Based Java (Brief overview)
- Conclusions
  - Benefits and limitations of constraints
  - Results from previous works
- References

# Constrained Conditional Models

- Extend the general conditional models with features, to satisfy certain constraints

# Constrained Conditional Models

- Extend the general conditional models with features, to satisfy certain constraints
- These constraints are a simpler way of encoding prior knowledge as compared to features

# Constrained Conditional Models

- Extend the general conditional models with features, to satisfy certain constraints
- These constraints are a simpler way of encoding prior knowledge as compared to features
- This is because features are less expressive, and in cases where certain rules need to consider the output  $Y$  in addition with the input  $X$ , constraints provide a better way

# Constrained Conditional Models

- Objective function :

$$f_{\phi,c}(x, y) = \sum_{i=1}^N w_i \phi_i(x, y) - \sum_{k=1}^M \rho_k d_{C_k}(x, y)$$

---

<sup>1</sup>Discussed in detail in [RR10]

# Constrained Conditional Models

- Objective function :

$$f_{\phi,c}(x, y) = \sum_{i=1}^N w_i \phi_i(x, y) - \sum_{k=1}^M \rho_k d_{C_k}(x, y)$$

- The second term, represents the effect of constraints on the objective function

---

<sup>1</sup>Discussed in detail in [RR10]

# Constrained Conditional Models

- Objective function :

$$f_{\phi,c}(x, y) = \sum_{i=1}^N w_i \phi_i(x, y) - \sum_{k=1}^M \rho_k d_{C_k}(x, y)$$

- The second term, represents the effect of constraints on the objective function
- $d_{C_k}(x)$  is 1 when the  $k^{\text{th}}$  constraint is violated; and 0 otherwise.

---

<sup>1</sup>Discussed in detail in [RR10]



# Constrained Conditional Models

- Objective function :

$$f_{\phi,c}(x, y) = \sum_{i=1}^N w_i \phi_i(x, y) - \sum_{k=1}^M \rho_k d_{C_k}(x, y)$$

- The second term, represents the effect of constraints on the objective function
- $d_{C_k}(x)$  is 1 when the  $k^{\text{th}}$  constraint is violated; and 0 otherwise.
- Hard constraints have the corresponding  $\rho_k$  as infinity; whereas, soft constraints have a real value for  $\rho_k$

---

<sup>1</sup>Discussed in detail in [RR10]

# Constrained Conditional Models

- Objective function :

$$f_{\phi,c}(x, y) = \sum_{i=1}^N w_i \phi_i(x, y) - \sum_{k=1}^M \rho_k d_{C_k}(x, y)$$

- The second term, represents the effect of constraints on the objective function
- $d_{C_k}(x)$  is 1 when the  $k^{th}$  constraint is violated; and 0 otherwise.
- Hard constraints have the corresponding  $\rho_k$  as infinity; whereas, soft constraints have a real value for  $\rho_k$
- The above objective function can be solved using integer linear programming approach <sup>1</sup>

---

<sup>1</sup>Discussed in detail in [RR10]

**There are two major approaches, that depend on how and when we use the constraints [CRRR08]:**

**There are two major approaches, that depend on how and when we use the constraints [CRRR08]:**

- *Learning plus Inference* : This method involves learning the feature vector, only with the features; that is, constraints are not even considered in the objective function. Once that is done, while testing we include the pre-learned constraint weights<sup>1</sup> along with the constraints in the objective function.

**There are two major approaches, that depend on how and when we use the constraints [CRRR08]:**

- *Learning plus Inference* : This method involves learning the feature vector, only with the features; that is, constraints are not even considered in the objective function. Once that is done, while testing we include the pre-learned constraint weights<sup>1</sup> along with the constraints in the objective function.
- *Inference Based Training* : In this approach we include the constraints term in the objective function as well. Note that this method itself can have two sub-approaches; which are (1) to learn constraint weights  $\rho$ 's with feature vector, and (2) to include pre-learned weights for constraints.

1 - Pre-learned constraints approach is generally used, as it is prior knowledge. However, for soft constraints, one may chose to learn the  $\rho$ 's also.

\* See table 1 in results for a comparison of approaches in semantic role labelling.

# Semi-supervised learning with Constraints

Constraint Driven Learning, referred from [RR10]; for semi supervised learning with constraints.

- Feedback plus constraints to label unlabelled examples
- Update the model via newly labelled data

## Input:

$N$ : learning cycles;  $U$ : unlabeled dataset;  $Tr = \{x, y\}$ : labeled training set;  $\{\rho_i\}$ : set of penalties;  $\{C_i\}$ : set of constraints;  $\gamma$ : balancing parameter with the supervised model;  $learn(Tr)$ : supervised learning algorithm;

1. Initialize  $w = w_0 = learn(Tr)$ .
2.  $w = w_0$ .
3. For  $N$  iterations do:
4.      $T = \phi$
5.     For each  $x \in U$
6.          $(x, \hat{y}) = \text{InferenceWithConstraints}(x, w, \{C_i\}, \{\rho_i\})$
7.          $T = T \cup \{(x, \hat{y})\}$
8.      $w = \gamma w_0 + (1 - \gamma) learn(T)$

# Research in CCMs for Natural Language Tasks

- Argument Labelling (or Semantic Role Labelling) - By Punyakanok et al '04 [PRYZ04]; Punyakanok, Roth, Yih '05 [KPRY05]
- Transliteration Discovery - By Chang, Goldwasser, Roth and Tu '09 [CGRT09]
- Semantic Parsing - By Clarke, Goldwasser, Chang and Roth '10 [CGCR10]
- Taxonomic Relation Classification - By Do and Roth '10 [DR10]
- Sentence Compression - By Clarke and Lapata '08 [CL08]
- Finding Best Antecedent - By Jindal and Roth '12 [JR12]
- Entity Matching - By Shen Li Doan '05 [SLD05]
- POS tagging - By Stokman '11 [Sto11]

## **Semantic Role Labelling**[PRYZ04]

The argument labels have been referred from PropBank[PGK05].



## Semantic Role Labelling[PRYZ04]

The argument labels have been referred from PropBank[PGK05].

- 1 Arguments cannot cover the predicate except those that contain only the verb or the verb and the following word

## Semantic Role Labelling[PRYZ04]

The argument labels have been referred from PropBank[PGK05].

- ① Arguments cannot cover the predicate except those that contain only the verb or the verb and the following word  
Ex. A verb cannot act as its own argument

## Semantic Role Labelling[PRYZ04]

The argument labels have been referred from PropBank[PGK05].

- 1 Arguments cannot cover the predicate except those that contain only the verb or the verb and the following word  
Ex. A verb cannot act as its own argument
- 2 Arguments cannot overlap with the clauses (they can be embedded in one another)

## Semantic Role Labelling[PRYZ04]

The argument labels have been referred from PropBank[PGK05].

- 1 Arguments cannot cover the predicate except those that contain only the verb or the verb and the following word  
Ex. A verb cannot act as its own argument
- 2 Arguments cannot overlap with the clauses (they can be embedded in one another)

Let  $C_{i,j}$  denote that the clause  $C_i$  and argument 'j' have words in common

Expression:  $\sum_i C_{i,j} \leq 1$

## Semantic Role Labelling (contd.)

## Semantic Role Labelling (contd.)

- ③ If a predicate is outside a clause, its arguments cannot be embedded in that clause

## Semantic Role Labelling (contd.)

- 3 If a predicate is outside a clause, its arguments cannot be embedded in that clause

A clause is independent with respect to other for its arguments. If  $A1$  is an argument of  $V$ , then:

Expression:  $C_{i,V} \geq C_{i,A1}$

## Semantic Role Labelling (contd.)

- ③ If a predicate is outside a clause, its arguments cannot be embedded in that clause

A clause is independent with respect to other for its arguments. If  $A_1$  is an argument of  $V$ , then:

Expression:  $C_{i,V} \geq C_{i,A_1}$

- ④ No overlapping or embedding arguments  
Two arguments are disjoint, which also means a single word can be a part of only one argument



## Semantic Role Labelling (contd.)

- ③ If a predicate is outside a clause, its arguments cannot be embedded in that clause

A clause is independent with respect to other for its arguments. If  $A1$  is an argument of  $V$ , then:

Expression:  $C_{i,V} \geq C_{i,A1}$

- ④ No overlapping or embedding arguments

Two arguments are disjoint, which also means a single word can be a part of only one argument

Expression :  $\sum_{i=1}^k z_{j_i,\phi} = k - 1$

## Semantic Role Labelling (contd.)

- ⑤ No duplicate argument classes for A0-A5,V

---

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'

## Semantic Role Labelling (contd.)

- 5 No duplicate argument classes for A0-A5,V  
Ex. A sentence cannot have two actors

---

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'

## Semantic Role Labelling (contd.)

- ⑤ No duplicate argument classes for A0-A5,V

Ex. A sentence cannot have two actors

$$\text{Expression}^2 : \sum_{i=1}^k z_{i,A0} \leq 1$$

---

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'

## Semantic Role Labelling (contd.)

- 5 No duplicate argument classes for A0-A5,V

Ex. A sentence cannot have two actors

$$\text{Expression}^2 : \sum_{i=1}^k z_{i,A0} \leq 1$$

- 6 Exactly one V argument per verb

---

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'

## Semantic Role Labelling (contd.)

- 5 No duplicate argument classes for A0-A5,V

Ex. A sentence cannot have two actors

$$\text{Expression}^2 : \sum_{i=1}^k z_{i,A0} \leq 1$$

- 6 Exactly one V argument per verb

There is only one 'main' verb that gets the V argument type

---

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'

## Semantic Role Labelling (contd.)

- 5 No duplicate argument classes for A0-A5,V

Ex. A sentence cannot have two actors

$$\text{Expression}^2 : \sum_{i=1}^k z_{i,A0} \leq 1$$

- 6 Exactly one V argument per verb

There is only one 'main' verb that gets the V argument type

$$\text{Expression} : \sum_{i=1}^k z_{i,V} = 1$$

---

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'

## Semantic Role Labelling (contd.)

- 5 No duplicate argument classes for A0-A5,V

Ex. A sentence cannot have two actors

$$\text{Expression}^2 : \sum_{i=1}^k z_{i,A0} \leq 1$$

- 6 Exactly one V argument per verb

There is only one 'main' verb that gets the V argument type

$$\text{Expression} : \sum_{i=1}^k z_{i,V} = 1$$

- 7 If there is C-V, then there should be a sequence of consecutive V, A1, and C-V pattern.

---

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'



## Semantic Role Labelling (contd.)

- 5 No duplicate argument classes for A0-A5,V

Ex. A sentence cannot have two actors

$$\text{Expression}^2 : \sum_{i=1}^k z_{i,A0} \leq 1$$

- 6 Exactly one V argument per verb

There is only one 'main' verb that gets the V argument type

$$\text{Expression} : \sum_{i=1}^k z_{i,V} = 1$$

- 7 If there is C-V, then there should be a sequence of consecutive V, A1, and C-V pattern.

Ex. When split is the verb in 'split it up', the A1 argument is 'it' and C-V argument is 'up'

---

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'

## Semantic Role Labelling (contd.)

- 5 No duplicate argument classes for A0-A5,V

Ex. A sentence cannot have two actors

$$\text{Expression}^2 : \sum_{i=1}^k z_{i,A0} \leq 1$$

- 6 Exactly one V argument per verb

There is only one 'main' verb that gets the V argument type

$$\text{Expression} : \sum_{i=1}^k z_{i,V} = 1$$

- 7 If there is C-V, then there should be a sequence of consecutive V, A1, and C-V pattern.

Ex. When split is the verb in 'split it up', the A1 argument is 'it' and C-V argument is 'up' Consider that  $j_1, j_2, j_3$  are consecutive arguments; and  $j_3$  is C-V. Then :

$$\text{Expression: } z_{j_3,C-V} \geq z_{j_1,A1} \text{ and } z_{j_3,C-V} \geq z_{j_2,A1}$$

<sup>2</sup> $z_i$  has been used to avoid double summation. Only one word will have argument 'i'

## Semantic Role Labelling (contd.)

---

<sup>3</sup>If there are  $\gamma$  reference pairs, then  $\gamma M$  inequalities are needed

## Semantic Role Labelling (contd.)

- 8 If there is an R-XXX argument, then there has to be an XXX argument

[A1 The pearls] [ R-A1 which ] [A0 I] [V left], [A2 to my daughter-in-law] are fake

---

<sup>3</sup>If there are  $\gamma$  reference pairs, then  $\gamma M$  inequalities are needed

## Semantic Role Labelling (contd.)

- 8 If there is an R-XXX argument, then there has to be an XXX argument

[A1 The pearls] [ R-A1 which ] [A0 I] [V left], [A2 to my daughter-in-law] are fake

Expression:  $\forall m \in \{1, 2, \dots, M\} : \sum_{i=1}^M z_{i,A0} \geq z_{m,R-A0}$ <sup>3</sup>

---

<sup>3</sup>If there are  $\gamma$  reference pairs, then  $\gamma M$  inequalities are needed

## Semantic Role Labelling (contd.)

- 8 If there is an R-XXX argument, then there has to be an XXX argument

[A1 The pearls] [ R-A1 which ] [A0 I] [V left], [A2 to my daughter-in-law] are fake

Expression:  $\forall m \in \{1, 2, \dots, M\} : \sum_{i=1}^M z_{i,A0} \geq z_{m,R-A0}$ <sup>3</sup>

- 9 If there is a C-XXX argument, then there has to be an XXX argument and C-XXX argument must occur after XXX  
[A1 The pearls], [A0 I] [V said], [C-A1 were left to my daughter-in-law]

---

<sup>3</sup>If there are  $\gamma$  reference pairs, then  $\gamma M$  inequalities are needed

## Semantic Role Labelling (contd.)

- 8 If there is an R-XXX argument, then there has to be an XXX argument

[A1 The pearls] [ R-A1 which ] [A0 I] [V left], [A2 to my daughter-in-law] are fake

Expression:  $\forall m \in \{1, 2, \dots, M\} : \sum_{i=1}^M z_{i,A0} \geq z_{m,R-A0}$ <sup>3</sup>

- 9 If there is a C-XXX argument, then there has to be an XXX argument and C-XXX argument must occur after XXX
- [A1 The pearls], [A0 I] [V said], [C-A1 were left to my daughter-in-law]

Expression:  $\forall m \in \{2, 3, \dots, M\} : \sum_{i=1}^{m-1} z_{i,A0} \geq z_{m,C-A0}$

---

<sup>3</sup>If there are  $\gamma$  reference pairs, then  $\gamma M$  inequalities are needed

## Semantic Role Labelling (contd.)



## Semantic Role Labelling (contd.)

- ⑩ Given the predicate, some argument classes are illegal

## Semantic Role Labelling (contd.)

- ⑩ Given the predicate, some argument classes are illegal  
Ex. Predicate 'stalk' can take only A0 or A1.(From PropBank frames). This constraint is encoded by summing up corresponding argument labels to zero

## Semantic Role Labelling (contd.)

- ⑩ Given the predicate, some argument classes are illegal  
Ex. Predicate 'stalk' can take only A0 or A1.(From PropBank frames). This constraint is encoded by summing up corresponding argument labels to zero

$$\text{Expression: } \sum_{i=1}^M z_{i,A5} = 0$$

## Transliteration Discovery [CGRT09]

## Transliteration Discovery [CGRT09]

- General Constraints
  - ① *Coverage*: Every character (sequence) must be mapped to a single character (sequence) or a blank character.

## Transliteration Discovery [CGRT09]

- General Constraints

- ① *Coverage*: Every character (sequence) must be mapped to a single character (sequence) or a blank character.  
Ex. The set devnagari symbol for 'ra' can be mapped to character set 'RA' in RAM; however it cannot map to any character from the next set of 'M'

## Transliteration Discovery [CGRT09]

- General Constraints

- ① *Coverage*: Every character (sequence) must be mapped to a single character (sequence) or a blank character.

Ex. The set devnagari symbol for 'ra' can be mapped to character set 'RA' in RAM; however it cannot map to any character from the next set of 'M'

Expression:  $\sum_i a_{ij} = 1$

## Transliteration Discovery [CGRT09]

- General Constraints

- 1 *Coverage*: Every character (sequence) must be mapped to a single character (sequence) or a blank character.

Ex. The set devnagari symbol for 'ra' can be mapped to character set 'RA' in RAM; however it cannot map to any character from the next set of 'M'

Expression:  $\sum_i a_{ij} = 1$

- 2 *No Crossing (Non-projectivity)*: The character mapping from source word to target word, should preserve the order of characters in the source language



## Transliteration Discovery [CGRT09]

- General Constraints

- 1 *Coverage*: Every character (sequence) must be mapped to a single character (sequence) or a blank character.

Ex. The set devnagari symbol for 'ra' can be mapped to character set 'RA' in RAM; however it cannot map to any character from the next set of 'M'

Expression:  $\sum_i a_{ij} = 1$

- 2 *No Crossing (Non-projectivity)*: The character mapping from source word to target word, should preserve the order of characters in the source language

$\forall i, j; a_{ij} = 1 \implies \forall l < i, \forall k > j; a_{lk} = 0$  and

$\forall i, j; a_{ij} = 1 \implies \forall l > i, \forall k < j; a_{lk} = 0$

## Transliteration Discovery

## Transliteration Discovery

- Language Specific Constraints
  - ① *Restricted mappings*: These include the phoneme(utterance) based constraints that a pair of languages imposes. Such valid mappings are maintained as prior knowledge, and used during the process of transliteration discovery.

## Transliteration Discovery

- Language Specific Constraints

- ① *Restricted mappings*: These include the phoneme(utterance) based constraints that a pair of languages imposes. Such valid mappings are maintained as prior knowledge, and used during the process of transliteration discovery. Expression: If  $\theta_t$  is the valid mapping for  $c_s$ , then any  $(c_s, c_t)$ , such that  $c_t \notin \theta_t$  is penalized.

## Transliteration Discovery

- Language Specific Constraints

- 1 *Restricted mappings*: These include the phoneme(utterance) based constraints that a pair of languages imposes. Such valid mappings are maintained as prior knowledge, and used during the process of transliteration discovery. Expression: If  $\theta_t$  is the valid mapping for  $c_s$ , then any  $(c_s, c_t)$ , such that  $c_t \notin \theta_t$  is penalized.
- 2 *Length Restriction*: This is an additional constraint that restricts the difference between lengths of two words.

## Transliteration Discovery

- Language Specific Constraints

- 1 *Restricted mappings*: These include the phoneme(utterance) based constraints that a pair of languages imposes. Such valid mappings are maintained as prior knowledge, and used during the process of transliteration discovery. Expression: If  $\theta_t$  is the valid mapping for  $c_s$ , then any  $(c_s, c_t)$ , such that  $c_t \notin \theta_t$  is penalized.
- 2 *Length Restriction*: This is an additional constraint that restricts the difference between lengths of two words. Expression: We fix a factor by which the length can vary as  $\gamma$ , then  
$$\forall v_s \in V_s, \forall v_t \in V_t, \text{ if } \gamma|v_s| > |v_t| \text{ or } \gamma|v_t| > |v_s|; \text{ then } \text{score}(F(v_s, v_t)) = \infty$$

# Examples of constraints as prior knowledge

## Semantic Parsing[CGCR10]

$\alpha_{cs}$  means, constituent  $c$  is bound with function  $s$ ; and  $\beta_{cs,dt}$  means that  $s(c)$  takes  $t(d)$  as argument

# Examples of constraints as prior knowledge

## Semantic Parsing[CGCR10]

$\alpha_{cs}$  means, constituent  $c$  is bound with function  $s$ ; and  $\beta_{cs,dt}$  means that  $s(c)$  takes  $t(d)$  as argument

- 1 A given constituent(word span) can be associated with exactly one logical symbol (function)



# Examples of constraints as prior knowledge

## Semantic Parsing[CGCR10]

$\alpha_{cs}$  means, constituent  $c$  is bound with function  $s$ ; and  $\beta_{cs,dt}$  means that  $s(c)$  takes  $t(d)$  as argument

- 1 A given constituent(word span) can be associated with exactly one logical symbol (function)

Ex. In 'the capital of India', the constituent India can be bound to only one function, which is ' $\lambda x.capital(x)$ '

# Examples of constraints as prior knowledge

## Semantic Parsing[CGCR10]

$\alpha_{cs}$  means, constituent  $c$  is bound with function  $s$ ; and  $\beta_{cs,dt}$  means that  $s(c)$  takes  $t(d)$  as argument

- 1 A given constituent(word span) can be associated with exactly one logical symbol (function)

Ex. In 'the capital of India', the constituent India can be bound to only one function, which is ' $\lambda x. capital(x)$ '

Expression:

$$\sum_i \alpha_{cs_i} = 1$$

# Examples of constraints as prior knowledge

## Semantic Parsing[CGCR10]

$\alpha_{cs}$  means, constituent  $c$  is bound with function  $s$ ; and  $\beta_{cs,dt}$  means that  $s(c)$  takes  $t(d)$  as argument

- 1 A given constituent(word span) can be associated with exactly one logical symbol (function)

Ex. In 'the capital of India', the constituent India can be bound to only one function, which is ' $\lambda x. capital(x)$ '

Expression:

$$\sum_i \alpha_{cs_i} = 1$$

- 2  $\beta_{cs,dt}$  is active if and only if  $\alpha_{cs}$  and  $\alpha_{dt}$  are active

# Examples of constraints as prior knowledge

## Semantic Parsing[CGCR10]

$\alpha_{cs}$  means, constituent  $c$  is bound with function  $s$ ; and  $\beta_{cs,dt}$  means that  $s(c)$  takes  $t(d)$  as argument

- 1 A given constituent(word span) can be associated with exactly one logical symbol (function)

Ex. In 'the capital of India', the constituent India can be bound to only one function, which is ' $\lambda x. capital(x)$ '

Expression:

$$\sum_i \alpha_{cs_i} = 1$$

- 2  $\beta_{cs,dt}$  is active if and only if  $\alpha_{cs}$  and  $\alpha_{dt}$  are active  
Ex. 'largest(next\_to(India))' exists only when next\_to(India) and largest(x) exist

# Examples of constraints as prior knowledge

## Semantic Parsing[CGCR10]

$\alpha_{cs}$  means, constituent  $c$  is bound with function  $s$ ; and  $\beta_{cs,dt}$  means that  $s(c)$  takes  $t(d)$  as argument

- 1 A given constituent(word span) can be associated with exactly one logical symbol (function)

Ex. In 'the capital of India', the constituent India can be bound to only one function, which is ' $\lambda x. capital(x)$ '

Expression:

$$\sum_i \alpha_{cs_i} = 1$$

- 2  $\beta_{cs,dt}$  is active if and only if  $\alpha_{cs}$  and  $\alpha_{dt}$  are active

Ex. 'largest(next\_to(India))' exists only when next\_to(India) and largest(x) exist

Expression:

$$\beta_{cs,dt} \leq \alpha_{cs} + \alpha_{dt} - 1$$

# Examples of constraints as prior knowledge

## Semantic Parsing (contd)

## Semantic Parsing (contd)

- ③ If  $\beta_{CS,dt}$  is active then  $s$  must be a function and types of  $s$  and  $t$  must be consistent

## Semantic Parsing (contd)

- ③ If  $\beta_{cs,dt}$  is active then s must be a function and types of s and t must be consistent

Ex. In 'largest(next\_to(India))' type of next\_to is 'entity to entity', and similarly for 'largest'



## Semantic Parsing (contd)

- ③ If  $\beta_{cs,dt}$  is active then  $s$  must be a function and types of  $s$  and  $t$  must be consistent

Ex. In 'largest(next\_to(India))' type of next\_to is 'entity to entity', and similarly for 'largest'

Expression: If we denote the co-domain(return type) of 't' as  $\text{cod}(t)$ , and domain of 's' as  $\text{dom}(s)$ ; then :

$$\beta_{cs,dt} \implies \text{dom}(s) = \text{cod}(t)$$

## Semantic Parsing (contd)

- ③ If  $\beta_{cs,dt}$  is active then  $s$  must be a function and types of  $s$  and  $t$  must be consistent

Ex. In 'largest(next\_to(India))' type of next\_to is 'entity to entity', and similarly for 'largest'

Expression: If we denote the co-domain(return type) of 't' as  $\text{cod}(t)$ , and domain of 's' as  $\text{dom}(s)$ ; then :

$$\beta_{cs,dt} \implies \text{dom}(s) = \text{cod}(t)$$

- ④ Functional composition is directional and acyclic

## Semantic Parsing (contd)

- ③ If  $\beta_{cs,dt}$  is active then  $s$  must be a function and types of  $s$  and  $t$  must be consistent

Ex. In 'largest(next\_to(India))' type of next\_to is 'entity to entity', and similarly for 'largest'

Expression: If we denote the co-domain(return type) of 't' as  $\text{cod}(t)$ , and domain of 's' as  $\text{dom}(s)$ ; then :

$$\beta_{cs,dt} \implies \text{dom}(s) = \text{cod}(t)$$

- ④ Functional composition is directional and acyclic  
Expression:

$$\begin{aligned} & \text{if } \beta_{c_i s_i, c_{(i+1)}, s_{(i+1)}} \forall i \in \{1, \dots, (n-1)\} \\ & \text{then; } c_i \neq c_j \forall j \in \{1, \dots, n\} - \{i\} \end{aligned}$$

## Sentence Compression[CL08]

- *Validity Constraints*: Constraints that ensure valid combinations of words are chosen

## Sentence Compression[CL08]

- *Validity Constraints*: Constraints that ensure valid combinations of words are chosen

Consider  $\alpha_i$  is 1 if  $x_i$  starts the sentence, then the constraint, 'only one word can begin a sentence', will be :  $\sum_{i=1}^n \alpha_i = 1$

## Sentence Compression[CL08]

- *Validity Constraints*: Constraints that ensure valid combinations of words are chosen

Consider  $\alpha_i$  is 1 if  $x_i$  starts the sentence, then the constraint, 'only one word can begin a sentence', will be :  $\sum_{i=1}^n \alpha_i = 1$

- *Modifier Constraints*: Relationships between headwords and their modifiers remain grammatical in compression

## Sentence Compression[CL08]

- *Validity Constraints*: Constraints that ensure valid combinations of words are chosen

Consider  $\alpha_i$  is 1 if  $x_i$  starts the sentence, then the constraint, 'only one word can begin a sentence', will be :  $\sum_{i=1}^n \alpha_i = 1$

- *Modifier Constraints*: Relationships between headwords and their modifiers remain grammatical in compression

Ex. If a non-clausal modifier is included in the compression, then head of the modifier must also be included

$\delta_i - \delta_j \geq 0$  ,  $\forall i, j$  ;  $x_j \in x_i$ 's 'ncmod's (non-clausal modifiers)

## Sentence Compression

- *Argument Structure Constraints:*



## Sentence Compression

- *Argument Structure Constraints:*
  - 1 If a verb is present in the compression, then so must be its arguments and vice versa

## Sentence Compression

- *Argument Structure Constraints:*

- 1 If a verb is present in the compression, then so must be its arguments and vice versa

$$\delta_i - \delta_j = 0, \forall i, j; x_j \in x_i\text{'s subjects/objects}$$

## Sentence Compression

- *Argument Structure Constraints:*

- 1 If a verb is present in the compression, then so must be its arguments and vice versa

$$\delta_i - \delta_j = 0, \forall i, j; x_j \in x_i \text{'s subjects/objects}$$

- 2 If the source sentence contains a verb, then compression must contain atleast one verb

## Sentence Compression

- *Argument Structure Constraints:*

- 1 If a verb is present in the compression, then so must be its arguments and vice versa

$$\delta_i - \delta_j = 0, \forall i, j; x_j \in x_i \text{'s subjects/objects}$$

- 2 If the source sentence contains a verb, then compression must contain atleast one verb

$$\sum_{i: x_i \in \text{verbs}} \delta_i \geq 1$$

# Examples of constraints as prior knowledge

**Finding best antecedent or Co-reference resolution**[JR12]

## Finding best antecedent or Co-reference resolution [JR12]

- *Modifier Constraint*: Two mentions should not have incompatible modifiers, like (small,large), etc.

## Finding best antecedent or Co-reference resolution [JR12]

- *Modifier Constraint*: Two mentions should not have incompatible modifiers, like (small,large), etc.  
Expression: If we maintain a pairwise list of disallowed combinations in  $S'_{mod}$ ; then :

$$x_i, x_j \in \text{modifiers}(x_k); (x_i, x_j) \in S'_{mod} \implies f = -\infty$$

## Finding best antecedent or Co-reference resolution [JR12]

- *Modifier Constraint*: Two mentions should not have incompatible modifiers, like (small,large), etc.  
Expression: If we maintain a pairwise list of disallowed combinations in  $S'_{mod}$ ; then :

$$x_i, x_j \in modifiers(x_k); (x_i, x_j) \in S'_{mod} \implies f = -\infty$$

- *Popular head constraint*: If some terms(heads) occur very often in a dataset, then mentions having same heads are considered co-referential, only if they remain to be so without considering the popular head.



## Finding best antecedent or Co-reference resolution [JR12]

- *Modifier Constraint*: Two mentions should not have incompatible modifiers, like (small,large), etc.  
Expression: If we maintain a pairwise list of disallowed combinations in  $S'_{mod}$ ; then :

$$x_i, x_j \in \text{modifiers}(x_k); (x_i, x_j) \in S'_{mod} \implies f = -\infty$$

- *Popular head constraint*: If some terms(heads) occur very often in a dataset, then mentions having same heads are considered co-referential, only if they remain to be so without considering the popular head.
- *Negation Constraint*: None of the two mentions should be present in a negated form.

**Finding best antecedent or Co-reference resolution - Domain specific constraints**

## Finding best antecedent or Co-reference resolution - Domain specific constraints

- *Body Parts Constraint*: If body parts (like chest, arm, head) are specified, they should not be incompatible

## Finding best antecedent or Co-reference resolution - Domain specific constraints

- *Body Parts Constraint*: If body parts (like chest, arm, head) are specified, they should not be incompatible
- *Anatomical Terms Constraint*: If anatomical terms (like proximal, anterior, dorsal) are specified, they should not be incompatible

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n + 1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n + 1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

- Sentence starts with a special start tag and ends with a special end tag



# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n + 1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

- Sentence starts with a special start tag and ends with a special end tag

$$\alpha_{0,start} = 1 \text{ and } \alpha_{(n+1),end} = 1$$

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n + 1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

- Sentence starts with a special start tag and ends with a special end tag  
 $\alpha_{0,start} = 1$  and  $\alpha_{(n+1),end} = 1$
- Tag  $i$  never precedes tag  $j$

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n + 1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

- Sentence starts with a special start tag and ends with a special end tag  
 $\alpha_{0,start} = 1$  and  $\alpha_{(n+1),end} = 1$
- Tag  $i$  never precedes tag  $j$   
Ex. Adjective never precedes a verb

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n+1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

- Sentence starts with a special start tag and ends with a special end tag

$$\alpha_{0,start} = 1 \text{ and } \alpha_{(n+1),end} = 1$$

- Tag  $i$  never precedes tag  $j$

Ex. Adjective never precedes a verb

$$\alpha_{k,i} + \alpha_{(k+1),j} \leq 1$$

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n+1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

- Sentence starts with a special start tag and ends with a special end tag

$$\alpha_{0,start} = 1 \text{ and } \alpha_{(n+1),end} = 1$$

- Tag  $i$  never precedes tag  $j$

Ex. Adjective never precedes a verb

$$\alpha_{k,i} + \alpha_{(k+1),j} \leq 1$$

- Presence of tag  $i$  implies presence of tag  $j$

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n+1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

- Sentence starts with a special start tag and ends with a special end tag

$$\alpha_{0,start} = 1 \text{ and } \alpha_{(n+1),end} = 1$$

- Tag  $i$  never precedes tag  $j$

Ex. Adjective never precedes a verb

$$\alpha_{k,i} + \alpha_{(k+1),j} \leq 1$$

- Presence of tag  $i$  implies presence of tag  $j$

Ex. Adverb and adjective imply a verb and a noun/pronoun respectively. i.e.

# Examples of constraints as prior knowledge

## Part Of Speech tagging[Sto11]

- There is exactly one tag for each token

$$\forall i \in 0, \dots, n+1 \sum_{j=1}^{j=N_t} \alpha_{i,j} = 1$$

- Sentence starts with a special start tag and ends with a special end tag

$$\alpha_{0,start} = 1 \text{ and } \alpha_{(n+1),end} = 1$$

- Tag  $i$  never precedes tag  $j$

Ex. Adjective never precedes a verb

$$\alpha_{k,i} + \alpha_{(k+1),j} \leq 1$$

- Presence of tag  $i$  implies presence of tag  $j$

Ex. Adverb and adjective imply a verb and a noun/pronoun respectively. i.e.  $\exists k, \alpha_{k,i} \implies \exists l, \alpha_{l,j}$

## Some other examples



## Some other examples

- State transitions occur only on punctuation marks. E.g. in addresses, etc.

## Some other examples

- State transitions occur only on punctuation marks. E.g. in addresses, etc.
- Constraints enforced by a relation on entities  
Ex. In the relation 'A born\_in B', A has to be living being, and B has to be a location

## Some other examples

- State transitions occur only on punctuation marks. E.g. in addresses, etc.
- Constraints enforced by a relation on entities  
Ex. In the relation 'A born\_in B', A has to be living being, and B has to be a location
- Unknown words that begin with capital letters (even when not mentioned in beginning of the sentence), are likely to be nouns

# Learning Based Java(LBJ)

- Learning Based Java is a frame work proposed by Rizzolo and Roth in '10, through [RR10]

# Learning Based Java(LBJ)

- Learning Based Java is a frame work proposed by Rizzolo and Roth in '10, through [RR10]
- It provides a way to introduce constraints on top of features through programming

# Learning Based Java(LBJ)

- Learning Based Java is a frame work proposed by Rizzolo and Roth in '10, through [RR10]
- It provides a way to introduce constraints on top of features through programming
- It can model and also reuse previously modelled tasks. On combining; 'best' solution is inferred in case of contradictions. This property helps us string together many small language models to implement a solution for a complex tasks.

# Learning Based Java(LBJ)

- Learning Based Java is a frame work proposed by Rizzolo and Roth in '10, through [RR10]
- It provides a way to introduce constraints on top of features through programming
- It can model and also reuse previously modelled tasks. On combining; 'best' solution is inferred in case of contradictions. This property helps us string together many small language models to implement a solution for a complex tasks.
- For a single model, LBJ compiles the code, and creates a corresponding 'java' file for the model. This file can now be used by any program that we wish.

## Features

- On changing a constraint or a feature, or adding it; the whole model need not be learnt again. Only that particular segment of the learning phase, which is affected will be learnt again.



## Features

- On changing a constraint or a feature, or adding it; the whole model need not be learnt again. Only that particular segment of the learning phase, which is affected will be learnt again.
- Feature extraction and learning produce several intermediate representations of the data they process. The LBJ compiler automates these processes.

## Features

- On changing a constraint or a feature, or adding it; the whole model need not be learnt again. Only that particular segment of the learning phase, which is affected will be learnt again.
- Feature extraction and learning produce several intermediate representations of the data they process. The LBJ compiler automates these processes.
- Overall, LBJ models a user's program as a collection of locally defined experts whose decisions are combined to make them globally coherent.

## In practice

LBJ has already been used in many systems. The list, referred from [RR10]; is as follows:

- The LBJ POS tagger2 reports a competitive 96.6% accuracy on the standard Wall Street Journal corpus.
- The named entity recognizer of (Ratinov and Roth, 2009): non-local features, gazetteers, and wikipedia are all incorporated into a system that achieves 90.8 F1 on the CoNLL-2003 dataset.
- The co-reference resolution system of (Bengtson and Roth, 2008) on the ACE 2004 dataset, employing only a single learned classifier and a single constraint; performs very well.

## Advantages

- The system becomes more accurate.

---

<sup>4</sup>Most of the mentioned works have achieved an improved performance, Tables 1 and 2 in results contains some tabulated results

## Advantages

- The system becomes more accurate.
- Number of training samples required to train is reduced.

---

<sup>4</sup>Most of the mentioned works have achieved an improved performance, Tables 1 and 2 in results contains some tabulated results

## Advantages

- The system becomes more accurate.
- Number of training samples required to train is reduced.
- The system remains to be simple, and yet performs well; over some complex systems, purely based on features <sup>4</sup>

---

<sup>4</sup>Most of the mentioned works have achieved an improved performance, Tables 1 and 2 in results contains some tabulated results

## Advantages

- The system becomes more accurate.
- Number of training samples required to train is reduced.
- The system remains to be simple, and yet performs well; over some complex systems, purely based on features <sup>4</sup>

## Limitations

---

<sup>4</sup>Most of the mentioned works have achieved an improved performance, Tables 1 and 2 in results contains some tabulated results

## Advantages

- The system becomes more accurate.
- Number of training samples required to train is reduced.
- The system remains to be simple, and yet performs well; over some complex systems, purely based on features <sup>4</sup>

## Limitations

- Support for non-linear constraints

---

<sup>4</sup>Most of the mentioned works have achieved an improved performance, Tables 1 and 2 in results contains some tabulated results



## Advantages

- The system becomes more accurate.
- Number of training samples required to train is reduced.
- The system remains to be simple, and yet performs well; over some complex systems, purely based on features <sup>4</sup>

## Limitations

- Support for non-linear constraints
- Hard constraints might overshadow soft constraints during learning

---

<sup>4</sup>Most of the mentioned works have achieved an improved performance, Tables 1 and 2 in results contains some tabulated results

# Comparison of learning approaches

Constraints	CRF-ML	CRF-P	CRF-IBT	VP
None	66.46	69.14	69.14	58.15
+No Dup	67.10	69.74	N/A	64.33
+Candidates	71.78	73.64	N/A	74.17
+Arguments	71.71	73.71	N/A	74.02
+Verb Pos	71.72	73.78	N/A	74.03
+Disallow	71.94	73.91	69.82	74.49
Training Time (hrs)	48	38	145	0.8

Table 1. Results with different approaches of CCM[CRRR08]

CRF-ML - Maximum likelihood estimation with Learning plus Inference (L+I); CRF-P - Perceptron with (L+I); CRF-IBT - Perceptron with Inference Based Training (IBT); VP - Voted perceptron with L+I protocol.

Learning plus Inference(L+I) works better than the Inference Based Training; in the above case study of Semantic Role Labelling.

Also note that the blanks under CRF-IBT indicate the instances when it took too long to respond.




# Comparison of approaches with and without CCM

# labeled samples	Supervised		Semi-supervised	
	HMM	HMM <sup>CCM</sup>	HMM	HMM <sup>CCM</sup>
Citations				
5	58.48	71.64 (31.69 %)	64.55	77.65 (36.96 %)
10	63.37	75.44 (32.94 %)	69.86	81.51 (38.67 %)
20	70.78	81.15 (35.49 %)	75.35	85.11 (39.61 %)
300	86.69	93.92 (54.29 %)	87.89	94.32 (53.07 %)
Advertisements				
5	53.90	61.16 (15.74 %)	60.75	70.79 (25.58 %)
10	61.21	68.12 (17.80 %)	66.56	75.40 (26.42 %)
20	67.69	72.64 (15.32 %)	71.36	77.56 (21.63 %)
100	76.29	80.80 (19.02 %)	77.38	82.00 (20.40 %)

Table 2. Improvements in HMM when CCM is used[CRR12].

The above data clearly proves that adding constraints to the feature-based approach reduces the training data required to attain comparable accuracy.




# References I




-  J. Clarke, D. Goldwasser, M. Chang, and D. Roth.  
Driving semantic parsing from the world's response.  
In *CoNLL*, Jul 2010.
-  M. Chang, D. Goldwasser, D. Roth, and Y. Tu.  
Unsupervised constraint driven learning for transliteration  
discovery.  
In *NAACL*, May 2009.
-  J. Clarke and M. Lapata.  
Global inference for sentence compression: An integer linear  
programming approach.  
*Journal of Artificial Intelligence Research*, 31:399–429, Jan  
2008.

# References II

-  M. Chang, L. Ratinov, and D. Roth.  
Structured learning with constrained conditional models.  
*Machine Learning*, 88(3):399–431, Jun 2012.
-  M. Chang, L. Ratinov, N. Rizzolo, and D. Roth.  
Learning and inference with constraints.  
In *AAAI*, Jul 2008.
-  Q. Do and D. Roth.  
Constraints based taxonomic relation classification.  
In *EMNLP*, pages 1099–1109, Massachusetts, USA, Oct 2010.
-  P. Jindal and D. Roth.  
Using knowledge and constraints to find the best antecedent.  
In *COLING*, Dec 2012.

## References III

-  P. Koomen, V. Punyakanok, D. Roth, and W. Yih.  
Generalized inference with multiple semantic role labeling systems shared task paper.  
In Ido Dagan and Dan Gildea, editors, *CoNLL*, pages 181–184, 2005.
-  Martha Palmer, Daniel Gildea, and Paul Kingsbury.  
The proposition bank: An annotated corpus of semantic roles.  
*Comput. Linguist.*, 31(1):71–106, March 2005.
-  V. Punyakanok, D. Roth, W. Yih, and D. Zimak.  
Semantic role labeling via integer linear programming inference.  
In *COLING*, pages 1346–1352, Geneva, Switzerland, Aug 2004.

-  N. Rizzolo and D. Roth.  
Learning based java for rapid development of nlp systems.  
In *LREC, Valletta, Malta, May 2010*.
-  Warren Shen, Xin Li, and AnHai Doan.  
Constraint-based entity matching.  
In *Proceedings of the 20th national conference on Artificial intelligence - Volume 2, AAAI'05*, pages 862–867. AAAI Press, 2005.
-  Tim Stokman.  
Constrained conditional models for improving the accuracy of a sequence tagger, May 2011.

# Acknowledgements

I thank,

**Prof. Ganesh Ramakrishnan**

*For his valuable guidance*

and

**Ajay Nagesh**

*For his help for the literature survey*