HC-Search: A Learning Framework for Search-based Structured Prediction

Janardhan Rao (Jana) Doppa
School of EECS, Oregon State University
“The Chatrapati Shivaji Terminus (CST) was attacked by two gunmen, one of whom, Ajmal Kasob, was later caught alive by the police and identified by eyewitnesses. The attacks began around 21:30 when the two men entered the passenger hall and opened fire, using AK-47 rifles. The attackers killed 58 people and injured 104 others, their assault ending at about 22:45.”

- Wikipedia, 2008 Mumbai Attacks
"The Chatrapati Shivaji Terminus (CST) was attacked by two gunmen, one of whom, Ajmal Kasob, was later caught alive by the police and identified by eyewitnesses. The attacks began around 21:30 when the two men entered the passenger hall and opened fire, using AK-47 rifles. The attackers killed 58 people and injured 104 others, their assault ending at about 22:45."

- Wikipedia, 2008 Mumbai Attacks
‘The Chatrapati Shivaji Terminus (CST) was attacked by two gunmen, one of whom, Ajmal Kasob, was later caught alive by the police and identified by eyewitnesses. The attacks began around 21:30 when the two men entered the passenger hall and opened fire, using AK-47 rifles. The attackers killed 58 people and injured 104 others, their assault ending at about 22:45.

- Wikipedia, 2008 Mumbai Attacks
“The Chatrapati Shivaji Terminus (CST) was attacked by two gunmen, one of whom, Ajmal Kasob, was later caught alive by the police and identified by eyewitnesses. The attacks began around 21:30 when the two men entered the passenger hall and opened fire, using AK-47 rifles. The attackers killed 58 people and injured 104 others, their assault ending at about 22:45.

- Wikipedia, 2008 Mumbai Attacks
The Chatrapati Shivaji Terminus (CST) was attacked by two gunmen, one of whom, Ajmal Kasab, was later caught alive by the police and identified by eyewitnesses. The attacks began around 21:30 when the two men entered the passenger hall and opened fire, using AK-47 rifles. The attackers killed 58 people and injured 104 others, their assault ending at about 22:45.

- Wikipedia, 2008 Mumbai Attacks
Deep Reading Bridge: Building Blocks

- POS tagging, Parsing, Co-reference resolution, Semantic role labeling ..
  - Inputs and outputs are highly structured

- Studied under a sub-field of machine learning called “Structured Prediction”
  - Generalization of standard classification
  - Exponential no. of classes (e.g., all valid parse trees)
Outline

• Overview of “structured prediction”
• Overview of ‘HC-Search framework’
• Learning Algorithms
  ▶ Search space learning
  ▶ Heuristic learning
  ▶ Cost function learning
• Experiments and Results
Outline

• Overview of structured prediction

• Overview of **HC-Search framework**

• Learning Algorithms
  - Search space learning
  - Heuristic learning
  - Cost function learning

• Experiments and Results
Classification to Structured Prediction
Learning a Classifier

Example problem:

\[ X - \text{image of a face} \]
\[ Y \in \{\text{male, female}\} \]
Learning a Classifier

Training Data
\{ (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \}

Example problem:
X - image of a face
Y \in \{ \text{male, female} \}
Learning a Classifier

Training Data
\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}

Example problem:
\( X \) - image of a face
\( Y \in \{ \text{male, female} \} \)

Learning Algorithm

\[ F(X, \theta) \]
Learning for **Simple** Outputs

Training Data
\[ \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \]

Example problem:
- \( X \) - image of a face
- \( Y \in \{\text{male, female}\} \)

Learning Algorithm

\[ F(X, \theta) \]

\( \theta \)

Feature vector

Class label
Learning for Simple Outputs

Training Data
\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}

Example problem:
- \(X\) - image of a face
- \(Y \in \{\text{male, female}\}\)

Learning Algorithm
\[ F(X, \theta) \]

Logistic Regression
Support Vector Machines
K Nearest Neighbor
Decision Trees
Neural Networks

\(\theta\) feature vector

\(\theta\) class label
Learning for **Structured Outputs**

Training Data
\( \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \)

Learning Algorithm
\[ F(X, \theta) \]

Part-of-Speech Tagging

**English Sentence:**
"The cat ran"

**Part-of-Speech Sequence:**
\(<article> <noun> <verb>\)

\[ Y = \text{set of all possible POS tag sequences} \]

**Exponential !!**
Learning for **Structured Outputs**

Training Data
\{ (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \}

Learning Algorithm
\[ F(X, \theta) \]

\[ Y = \text{set of all valid parse trees} \]

**Dependency Parsing**

English Sentence:
"Red figures on the screen indicated falling stocks"

Dependency Tree:

- _ROOT_
- Red
- figures
- on
- the
- screen
- indicated
- falling
- stocks

Exponential !!
Learning for Structured Outputs

Training Data
\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}

Learning Algorithm
\[ \theta \]

F(X, \theta)

Co-reference Resolution

Text with input mentions:
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Co-reference Output:
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

\[ Y = \text{set of all possible clusterings} \]

Exponential !!
Learning for **Structured Outputs**

**Training Data**

\[ \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \]

**Handwriting Recognition**

**Handwritten Word:**

Structured

**Letter Sequence:**

Structured

\[ Y = \text{set of all possible letter sequences} \]

**Exponential !!**
Learning for Structured Outputs

Training Data
\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}

Learning Algorithm
\[ F(X, \theta) \]

\[ \theta \]

\[ Y = \text{set of all possible labelings} \]

Exponential !!
Structured Prediction: Approaches

• Most algorithms learn parameters of linear models
  \( \phi(x, y) \) is n-dim feature vector over input-output pairs
  \( w \) is n-dim parameter vector

\[
F(x) = \arg\min_{y \in Y} w \cdot \phi(x, y)
\]
**Structured Prediction: Approaches**

- Most algorithms learn parameters of linear models
  - $\phi(x, y)$ is $n$-dim feature vector over input-output pairs
  - $w$ is $n$-dim parameter vector

$$F(x) = \arg\min_{y \in Y} w \cdot \phi(x, y)$$

**Example: Part-of-Speech Tagging**

$x = \text{“The cat ran”}$  \hspace{1cm}  $y = <\text{article}> <\text{noun}> <\text{verb}>$

$\phi(x, y)$ may have unary and pairwise features

- **unary feature:** e.g. # of times ‘the’ is paired with $<\text{article}>$
- **pairwise feature:** e.g. # of times $<\text{article}>$ followed by $<\text{verb}>$
Structured Prediction: Approaches

• Most algorithms learn parameters of linear models
  \( \phi(x, y) \) is \( n \)-dim feature vector over input-output pairs
  \( w \) is \( n \)-dim parameter vector

\[
F(x) = \arg \min_{y \in Y} w \cdot \phi(x, y)
\]

• Most algorithms for simple outputs have been generalized to structured outputs, e.g.
  \( \text{SVMs} \Rightarrow \text{Structured SVM} \) [Tsochantaridis et al. 2004]
  \( \text{Logistic Regression} \Rightarrow \text{Conditional Random Fields} \) [Lafferty et al. 2001]
Key Challenge: “Argmin” Inference

\[ F(x) = \arg \min_{y \in Y} w \cdot \phi(x, y) \]

|Y| is generally exponentially large

- “Argmin” Inference:
  
  find the min. scoring output in an exponentially large set of possible outputs (e.g., all valid parse trees in dependency parsing)
Key Challenge: “Argmin” Inference

\[ F(x) = \arg \min_{y \in Y} w \cdot \phi(x, y) \]

|\( Y \) is generally exponentially large

- Most learning algorithms assume exact “Argmin” inference
  - Perform “Argmin” inference many times in inner loop
Key challenge: “Argmin” Inference

\[
F(x) = \arg \min_{y \in Y} w \cdot \phi(x, y)
\]

• Lot of work on doing this inference and learning with it over the last decade or so

• Time complexity of inference depends on the dependency structure of features \( \phi(x, y) \)
  ▲ Efficient inference algorithms exist only for simple features
  ▲ ... in practice use approximate inference and hope it works
Outline

• Overview of “structured prediction”

• Overview of ‘HC-Search framework’

• Learning Algorithms
  ▲ Search space learning
  ▲ Heuristic learning
  ▲ Cost function learning

• Experiments and Results
HC-Search framework: Overview

- Structured Prediction as a search process
HC-Search framework: Overview

- Structured Prediction as a search process
  - Generate high-quality outputs
  - Select from the generated outputs
HC-Search framework: Overview

• Structured Prediction as a search process

  • **Generate high-quality outputs**
    - Start from a reasonably good output (Initial solution)
HC-Search framework: Overview

• Structured Prediction as a search process

• Generate high-quality outputs
  o Start from a reasonably good output (Initial solution)
  o Search by making changes to the initial solution
HC-Search framework: Overview

- Structured Prediction as a search process

- Generate high-quality outputs
  - Start from a reasonably good output (Initial solution)
  - Search by making changes to the initial solution
  - Use a heuristic function $H$ to make the search efficient
HC-Search framework: Overview

• Structured Prediction as a search process

• **Generate high-quality outputs**
  - Start from a reasonably good output (Initial solution)
  - Search by making changes to the initial solution
  - Use a heuristic function $H$ to make the search efficient

• **Select from the generated outputs**
  - Use a cost function $C$ to score the potential outputs
Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Ground truth output (y*):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”
Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.
Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.

Initial Solution

Nodes = input-output pairs
Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.

Heuristic value 0.68

Barack Obama He
Hillary Clinton First Lady
Secretary of state Her she

Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.

Heuristic value 0.99

Barack Obama He
Hillary Clinton Her She First Lady
Secretary of state Barack Obama He
First Lady she

Initial Solution
Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.

Initial Solution

Heuristic value

0.68

0.99
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Barack Obama
He

Hillary Clinton
First Lady

Secretary of state
Her
she

Hillary Clinton
Secretary of state
Barack Obama
He

First Lady
Her
she

Secretary of state
Barack Obama
He

Hillary Clinton
Her
She
First Lady
"Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady."

Barack Obama  
He

Hillary Clinton  
First Lady

Secretary of state  
Her  
she

Heuristic value

0.98

"Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady."

Barack Obama  
He

Secretary of state  
First Lady  
her

Hillary Clinton  
she

0.32

"Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady."

Barack Obama  
He

Hillary Clinton  
Secretary of state  
Her  
She  
First Lady

Initial Solution
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Generated Outputs:
Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

**Initial Solution**

Barack Obama
- He

Hillary Clinton
- Her
- First Lady

Secretary of state
- Her
- She

Cost
- 0.52

Selected Output:
HC-Search: Properties

• **Anytime predictions**
  - Stop the search at any point and return the best cost output

• **Minimal restrictions on the complexity of heuristic and cost functions**
  - Only needs to be evaluated on complete input-output pairs
  - Can use higher-order features
Tradeoff Space

• Computational cost of inference
• Difficulty of the learning task(s)
• Constraints on feature design
Conditional Random Field / Structured SVM / Structured Perceptron

<table>
<thead>
<tr>
<th>Method</th>
<th>Cost of Inference</th>
<th>Difficulty of Learning</th>
<th>Design of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF/SSVM</td>
<td>Expensive; NP-Hard in many cases</td>
<td>Very difficult</td>
<td>Highly constrained</td>
</tr>
</tbody>
</table>

- One set of features and one learned function must do all of the work
- Learned function must “defend against” all incorrect solutions generated by the argmax inference
### Easy-First

<table>
<thead>
<tr>
<th>Method</th>
<th>Cost of Inference</th>
<th>Difficulty of Learning</th>
<th>Design of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF/SSVM</td>
<td>Expensive; NP-Hard in many cases</td>
<td>Very difficult</td>
<td>Highly constrained</td>
</tr>
<tr>
<td>Easy First</td>
<td>Cheap (greedy)</td>
<td>Moderate</td>
<td>Highly constrained</td>
</tr>
</tbody>
</table>

- One set of features and one learned function still do all of the work
- Learned function must “defend against” only those incorrect solutions encountered during greedy search
- Exploits the fact that correct answer can be reached by many different paths—only need to find one of them
## HC-Search

<table>
<thead>
<tr>
<th>Method</th>
<th>Cost of Inference</th>
<th>Difficulty of Learning</th>
<th>Design of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF/SSVM</td>
<td>Expensive; NP-Hard in many cases</td>
<td>Very difficult</td>
<td>Highly constrained</td>
</tr>
<tr>
<td>Easy First</td>
<td>Cheap (greedy)</td>
<td>Moderate</td>
<td>Highly constrained</td>
</tr>
<tr>
<td>HC-Search</td>
<td>Moderate</td>
<td>Easier</td>
<td>Less constrained</td>
</tr>
</tbody>
</table>

- Separate functions are learned for different roles
  - allows both local and global features to be used
  - $H$ function needs to generate good solutions at shallow depth. Error-tolerant
  - $C$ function needs to properly score only those solutions generated by $H$
HC-Search framework: Key Ingredients

• Structured Prediction as a search process

• Generate high-quality outputs
  - Start from a reasonably good output (Initial solution)
  - Search by making changes to the initial solution
  - Use a heuristic function $H$ to make the search efficient

• Select from the generated outputs
  - Use a cost function $C$ to score the potential outputs
Outline

• Overview of “structured prediction”
• Overview of ‘HC-Search framework’
• Learning Algorithms
  ▲ Search space learning
  ▲ Heuristic learning
  ▲ Cost function learning
• Experiments and Results
HC-Search: Search Space Learning

• **Objective:**
  
  High-quality outputs can be located at small depth
HC-Search: Search Space Learning

• **Objective:**
  - High-quality outputs can be located at small depth

• **Our Solution:**
  - Limited Discrepancy Search (LDS) Space [Doppa et al. 2012]
  - Defined in terms of a greedy predictor
HC-Search: Search Space Learning

• Objective:
  ▲ High-quality outputs can be located at small depth

• Our Solution:
  ▲ Limited Discrepancy Search (LDS) Space [Doppa et al. 2012]
  ▲ Defined in terms of a greedy predictor
HC-Search: Search Space Learning

• Greedy Predictor:

  Makes a sequence of greedy decisions to construct outputs
HC-Search: Search Space Learning

• Greedy Predictor:

  \[ \text{Makes a sequence of greedy decisions to construct outputs} \]

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”
HC-Search: Search Space Learning

• Greedy Predictor:
  ▲ Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”
Greedy Predictor:

Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

Barack Obama
Greedy Predictor:

Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

Barack Obama
HC-Search: Search Space Learning

• Greedy Predictor:
  ▲ Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

Barack Obama  Hillary Clinton
HC-Search: Search Space Learning

• Greedy Predictor:
  ▲ Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

Barack Obama  Hillary Clinton
HC-Search: Search Space Learning

• Greedy Predictor:

▲ Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

Barack Obama  Hillary Clinton  Secretary of state
HC-Search: Search Space Learning

• Greedy Predictor:
  - Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

- Barack Obama
- Hillary Clinton
- Secretary of state
HC-Search: Search Space Learning

• Greedy Predictor:
  ▲ Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

Barack Obama

He

Hillary Clinton

Secretary of state
HC-Search: Search Space Learning

• Greedy Predictor:
  ▲ Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

- Barack Obama
- Hillary Clinton
- Secretary of state
Greedy Predictor:

Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:

- Barack Obama
- He
- Hillary Clinton
- Secretary of state
- her
• **Greedy Predictor:**

  Makes a sequence of greedy decisions to construct outputs

**Text with input mentions (x):**

“**Barack Obama** nominated **Hillary Clinton** as his **secretary of state** on Monday. **He chose her** because **she** had foreign affair experience as a former **First Lady**.”

**Partial Output:**

Barack Obama  
He

Hillary Clinton

Secretary of state  
her
HC-Search: Search Space Learning

- Greedy Predictor:
  - Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Partial Output:
Greedy Predictor:

- Makes a sequence of greedy decisions to construct outputs

Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former [First Lady].”

Partial Output:
**Greedy Predictor:**

- Makes a sequence of greedy decisions to construct outputs

**Text with input mentions (x):**

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

**Predicted Output:**

- Barack Obama
- Hillary Clinton
- Secretary of state
- He
- First Lady
- her
- she
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”
"Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady."

Predicted Output:

- Barack Obama
- Hillary Clinton
- Secretory of state
  - He
  - First Lady
  - her
  - she

- Limited Discrepancy Search [Harvey and Ginsberg, 1995]

  - Key idea: correct the response of greedy predictor at a small no. of critical errors to produce high-quality outputs
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Predicted Output:

Partial output after correction:
Text with input mentions (x):

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Predicted Output:

Barack Obama
He

Hillary Clinton
First Lady

Secretary of state
her
she

Limited Discrepancy Search Illustration
"Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady."

Barack Obama
He

Hillary Clinton
First Lady

Secretary of state
Her
she

Initial Solution
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Barack Obama
He

Hillary Clinton
First Lady

Secretary of state
Her

...
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Hillary Clinton
First Lady

Secretary of state
Her

Barack Obama
He

One correction
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Hillary Clinton
He
First Lady

Secretary of state
Her
she

Barack Obama
He

Hillary Clinton
Her

Secretary of state
First Lady
she

Two corrections

Initial Solution

Barack Obama
He

Secretary of state
Her

She
First Lady

Hillary Clinton

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Hillary Clinton

Secretary of state
Barack Obama
He

She
First Lady
Outline

• Overview of “structured prediction”
• Overview of ‘HC-Search framework’
• Learning Algorithms
  ▲ Search space learning
  ▲ Heuristic learning
  ▲ Cost function learning
• Experiments and Results
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Barack Obama
He

Hillary Clinton
First Lady

Secretary of state
Her
she

Selected Output:
HC-Search: Loss Decomposition

"Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady."

Initial Solution

Best cost output
HC-Search: Loss Decomposition

Best cost output

Loss = 0.22
HC-Search: Loss Decomposition

Best cost output

Loss = 0.22

Min loss output

Loss = 0.09
HC-Search: Loss Decomposition

**Initial Solution**

- **Best cost output**
  - Loss = 0.22

- **Min loss output**
  - Loss = 0.09

**Overall loss** \( \epsilon = 0.22 \)

**Generation loss** \( \epsilon_H = 0.09 \)  
(Heuristic function)

**Selection loss** \( \epsilon_C = 0.22 - 0.09 \)  
(Cost function)
HC-Search: Loss Decomposition

\[ C(x, y) = w_c \cdot \phi(x, y) \]
\[ H(x, y) = w_H \cdot \phi(x, y) \]

\[ \epsilon = \epsilon_H + \epsilon_{C|H} \]

Overall expected loss
Generation loss (Heuristic function)
Selection loss (Cost function)
HC-Search: Learning

\[ \epsilon = \epsilon_H + \epsilon_{C|H} \]

Overall loss  Generation loss (Heuristic function)  Selection loss (Cost function)

- **Key idea:** Greedy stage-wise minimization guided by the loss decomposition
\[ \epsilon = \epsilon_H + \epsilon_{C|H} \]

**Overall loss**

**Generation loss**
(Heuristic function)

**Selection loss**
(Cost function)

- **Key idea:** Greedy stage-wise minimization guided by the loss decomposition

  - **Step 1:** \( \hat{H} = \arg\min_{H \in H} \epsilon_H \) (heuristic training)
**HC-Search: Learning**

\[ \epsilon = \epsilon_H + \epsilon_{C|H} \]

- **Overall loss**
- **Generation loss** (Heuristic function)
- **Selection loss** (Cost function)

**Key idea:** Greedy stage-wise minimization guided by the loss decomposition

- **Step 1:** \( \hat{H} = \arg \min_{H \in H} \epsilon_H \) (heuristic training)
- **Step 2:** \( \hat{C} = \arg \min_{C \in C} \epsilon_{C|\hat{H}} \) (cost function training)
HC-Search: Heuristic learning

• Learning Objective:
  ▲ Guide the search quickly towards high-quality (low loss) outputs
HC-Search: Heuristic Learning

• Given a search procedure (e.g., greedy search)

  **Key idea: Imitation of true loss function**

  - Conduct searches on training example using the true loss function as a heuristic
    (generally is a good way to produce good outputs)

  - Learn a heuristic function that tries to imitate the observed search behavior
"Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady."

Hillary Clinton
First Lady

Secretary of state
Her
she

Barack Obama
He

Initial Solution

True loss

0.65

0.23

0.39

0.17

0
“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

Initial Solution

True loss

0.23

0.39

0.65

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”

“Barack Obama nominated Hillary Clinton as his secretary of state on Monday. He chose her because she had foreign affair experience as a former First Lady.”
HC-Search: Heuristic Learning

Ranking examples

Can prove generalization bounds on learned heuristic

[Doppa et al., 2012]
Outline

• Overview of “structured prediction”

• Overview of ‘HC-Search framework’

• Learning Algorithms
  • Search space learning
  • Heuristic learning
  • Cost function learning

• Experiments and Results
HC-Search: Cost Function Learning

$$\epsilon = \epsilon_H + \epsilon_{C|H}$$

Overall loss
Generation loss (Heuristic function)
Selection loss (Cost function)

**Key idea:** Greedy stage-wise minimization guided by the loss decomposition

▲ Step 1: $$\hat{H} = \arg\min_{H \in H} \epsilon_H$$ (heuristic training)

▲ Step 2: $$\hat{C} = \arg\min_{C \in C} \epsilon_{C|\hat{H}}$$ (cost function training)
HC-Search: Cost Function Learning

**Learning Objective:**

Correctly score the outputs generated by the heuristic as per their losses

Set of all outputs generated by the learned heuristic $\hat{H}$
HC-Search: Cost function Learning

• Learning to Rank:

Best loss outputs

Non-best loss outputs

• Create a ranking example between every pair of outputs \( (y_{\text{best}}, y) \) such that: \( C(x, y_{\text{best}}) < C(x, y) \)
Cost function Learning

Ranking examples

Can borrow generalization bounds from rank-learning literature
[Agarwal and Roth, 2005 & Agarwal and Niyogi, 2009]
Outline

• Overview of “structured prediction”

• Overview of ‘HC-Search framework’

• Learning Algorithms
  ▲ Search space learning
  ▲ Heuristic learning
  ▲ Cost function learning

• Experiments and Results
  ▲ Benchmark structured prediction domains
  ▲ POS tagging and Chunking results
  ▲ Co-reference resolution results
Domains

- **Handwriting recognition** [Taskar et al., 2003]
  - *HW-Small* and *HW-Large*

  \[ x = \text{structured} \quad y = \text{structured} \]

- **NET-Talk** [Sejnowski and Rosenberg, 1987]
  - *Stress* and *Phoneme* prediction

  \[ x = \text{“photograph”} \quad y = /f-Ot@graf-/ \]

- **Scene labeling** [Vogel et al., 2007]

  \[ x = \quad y = \]
Results: comparison to state-of-the-art

Error-rates of different structured prediction algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>HW-Small</th>
<th>HW-Large</th>
<th>Phoneme</th>
<th>Scene labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>HC-Search</td>
<td>13.35</td>
<td>03.78</td>
<td>16.05</td>
<td>19.71</td>
</tr>
<tr>
<td>C-Search</td>
<td>17.41</td>
<td>07.41</td>
<td>20.91</td>
<td>27.05</td>
</tr>
<tr>
<td>CRF</td>
<td>19.97</td>
<td>13.11</td>
<td>21.09</td>
<td>-</td>
</tr>
<tr>
<td>SVM-Struct</td>
<td>19.64</td>
<td>12.49</td>
<td>21.70</td>
<td>-</td>
</tr>
<tr>
<td>Recurrent</td>
<td>34.33</td>
<td>25.13</td>
<td>26.42</td>
<td>43.36</td>
</tr>
<tr>
<td>SEARN</td>
<td>17.88</td>
<td>09.42</td>
<td>22.74</td>
<td>37.69</td>
</tr>
<tr>
<td>CASCADES</td>
<td>30.38</td>
<td>12.05</td>
<td>30.23</td>
<td>-</td>
</tr>
</tbody>
</table>

Outstanding Paper Award, AAAI 2013
Results: comparison to state-of-the-art

Error-rates of different structured prediction algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>HW-Small</th>
<th>HW-Large</th>
<th>Phoneme</th>
<th>Scene labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>HC-Search</td>
<td>13.35</td>
<td>03.78</td>
<td>16.05</td>
<td>19.71</td>
</tr>
<tr>
<td>C-Search</td>
<td>17.41</td>
<td>07.41</td>
<td>20.91</td>
<td>27.05</td>
</tr>
<tr>
<td>CRF</td>
<td>19.97</td>
<td>13.11</td>
<td>21.09</td>
<td>-</td>
</tr>
<tr>
<td>SVM-Struct</td>
<td>19.64</td>
<td>12.49</td>
<td>21.70</td>
<td>-</td>
</tr>
<tr>
<td>Recurrent</td>
<td>34.33</td>
<td>25.13</td>
<td>26.42</td>
<td>43.36</td>
</tr>
<tr>
<td>SEARN</td>
<td>17.88</td>
<td>09.42</td>
<td>22.74</td>
<td>37.69</td>
</tr>
<tr>
<td>CASCADES</td>
<td>30.38</td>
<td>12.05</td>
<td>30.23</td>
<td>-</td>
</tr>
</tbody>
</table>

- **HC-Search** outperforms all the other algorithms including C-Search (our prior approach that uses a single function C to serve the dual roles of heuristic and cost function)
Results: Loss Decomposition Analysis

\[ \epsilon = \epsilon_H + \epsilon_{C|H} \]

- Overall expected loss
- Generation loss (Heuristic function)
- Selection loss (Cost function)
# Results: Loss decomposition analysis

<table>
<thead>
<tr>
<th>Phonetic Symbol</th>
<th>Phoneme</th>
<th>Scene Labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR</strong></td>
<td>𝜖</td>
<td>𝜖</td>
</tr>
<tr>
<td></td>
<td>𝜖_H</td>
<td>𝜖_C</td>
</tr>
<tr>
<td><strong>HC-Search</strong></td>
<td>16.05</td>
<td>03.98</td>
</tr>
<tr>
<td></td>
<td><strong>12.07</strong></td>
<td>19.71</td>
</tr>
<tr>
<td></td>
<td>5.82</td>
<td>13.89</td>
</tr>
</tbody>
</table>
### Results: Loss decomposition analysis

<table>
<thead>
<tr>
<th></th>
<th>Phoneme</th>
<th></th>
<th>Scene labeling</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR</strong></td>
<td>$\epsilon$</td>
<td>$\epsilon_H$</td>
<td>$\epsilon_{C</td>
<td>H}$</td>
</tr>
<tr>
<td><strong>HC-Search</strong></td>
<td>16.05</td>
<td>03.98</td>
<td><strong>12.07</strong></td>
<td>19.71</td>
</tr>
</tbody>
</table>

- Selection loss $\epsilon_{C|H}$ contributes more to the overall loss
Results: Loss decomposition analysis

<table>
<thead>
<tr>
<th></th>
<th>Phoneme</th>
<th>Scene labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR</strong></td>
<td>$\epsilon$</td>
<td>$\epsilon_H$</td>
</tr>
<tr>
<td>HC-Search</td>
<td>16.05</td>
<td>03.98</td>
</tr>
<tr>
<td>C-Search</td>
<td>20.91</td>
<td>04.38</td>
</tr>
</tbody>
</table>
Results: Loss decomposition analysis

<table>
<thead>
<tr>
<th>ERROR</th>
<th>Phoneme</th>
<th>Scene labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\epsilon$</td>
<td>$\epsilon_H$</td>
</tr>
<tr>
<td>HC-Search</td>
<td>16.05</td>
<td>03.98</td>
</tr>
<tr>
<td>C-Search</td>
<td>20.91</td>
<td>04.38</td>
</tr>
</tbody>
</table>

- Improvement of HC-Search over C-Search is due to the improvement in the selection loss
## Results: Loss decomposition analysis

<table>
<thead>
<tr>
<th>Error</th>
<th>Phoneme</th>
<th>Scene labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\epsilon$</td>
<td>$\epsilon_H$</td>
</tr>
<tr>
<td>HC-Search</td>
<td>16.05</td>
<td>03.98</td>
</tr>
<tr>
<td>C-Search</td>
<td>20.91</td>
<td>04.38</td>
</tr>
</tbody>
</table>

- Improvement of HC-Search over C-Search is due to the improvement in the selection loss.
- Clearly shows the advantage of separating the roles of heuristic and cost function.
Outline

• Overview of “structured prediction”

• Overview of ‘HC-Search framework’

• Learning Algorithms
  ▲ Search space learning
  ▲ Heuristic learning
  ▲ Cost function learning

• Experiments and Results
  ▲ Benchmark structured prediction domains
  ▲ POS tagging and Chunking results
  ▲ Co-reference resolution results
NLP domains: POS tagging and Chunking

Error-rates of different structured prediction algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>POS tagging</th>
<th>Chunking</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-Search</td>
<td>96.95</td>
<td>94.73</td>
</tr>
<tr>
<td>CRF</td>
<td>96.84</td>
<td>94.77</td>
</tr>
<tr>
<td>SVM-Struct</td>
<td>96.81</td>
<td>93.64</td>
</tr>
<tr>
<td>Recurrent</td>
<td>92.15</td>
<td>88.51</td>
</tr>
<tr>
<td>SEARN</td>
<td>95.83</td>
<td>94.44</td>
</tr>
<tr>
<td>CASCADES</td>
<td>96.82</td>
<td>-</td>
</tr>
</tbody>
</table>
Outline

• Overview of “structured prediction”
• Overview of ‘HC-Search framework’
• Learning Algorithms
  ▲ Search space learning
  ▲ Heuristic learning
  ▲ Cost function learning
• Experiments and Results
  ▲ Benchmark structured prediction domains
  ▲ POS tagging and Chunking results
  ▲ Co-reference resolution results
NLP domains: Co-reference Resolution

• Ongoing work:

  ▶ Started very recently

  ▶ Learned only greedy predictor so far
NLP domains: Co-reference Results

- **Dataset:**
  - ACE2004 Multi-Linguistic Training Corpus

- **Experimental Setup:**
  - 268 documents for training, 68 docs for validation, and 107 docs for testing
  - Standard gold mentions
  - Same features as Stoyanov et al., 2012
  - SVM-Rank and LambdaMART as base learners. Hyper-parameters are tuned via validation set.

- **Baselines:**
  - Easy-first (Stoyanov et al., 2012)
  - Threshold for “halting” was tuned via validation set
## Co-reference Results: Greedy Predictor

<table>
<thead>
<tr>
<th></th>
<th>MUC</th>
<th>B-Cubed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Greedy (SVM-Rank)</td>
<td>74.37</td>
<td>71.85</td>
</tr>
<tr>
<td>Greedy (LambdaMART)</td>
<td>82.75</td>
<td>69.25</td>
</tr>
<tr>
<td>Easy-First</td>
<td>85.04</td>
<td>57.75</td>
</tr>
</tbody>
</table>

- Results with greedy predictor are competitive with the state-of-the-art
- HC-Search with LDS space (by employing greedy predictor) can improve the results further – *ongoing work*
Summary and Take Home Message

• HC-Search Framework
  - Generate high-quality outputs and select the best from them
  - **Division of work:** Initial solution, heuristic and cost function
  - Each function has an easier task, can employ different features

• Take Home Message
  - Viewing structured prediction as a “search problem” can achieve state-of-the-art performance
  - When applying to new problems:
    - Design a “high-quality” search space (e.g., LDS space)
    - Choose a suitable search procedure
    - Use HC-Search “loss decomposition” to train and debug
Questions ?