S-MART: Novel Tree-based Structured Learning Algorithms Applied to Tweet Entity Linking

Yi Yang* and Ming-Wei Chang#

*Georgia Institute of Technology, Atlanta
#Microsoft Research, Redmond
Traditional NLP Settings

- High dimensional sparse features (e.g., lexical features)
  - Languages are naturally in high dimensional spaces.
  - Powerful! Very expressive.

- Linear models
  - Linear Support Vector Machine
  - Maximize Entropy model

Sparse features + Linear models
Rise of Dense Features

- Low dimensional embedding features
- Low dimensional statistics features

Dense features + Non-linear models
Non-linear Models

- Neural networks
Non-linear Models

- Neural networks
- Kernel methods
- Tree-based models (e.g., Random Forest, Boosted Tree)
Non-linear Models

- Neural networks
- Kernel methods
- Tree-based models (e.g., Random Forest, Boosted Tree)

![Bar chart showing the number of papers in ACL'15 for different model types.]

- **Neural**
  - Number: 35
- **Kernel**
  - Number: 5
- **Tree**
  - Number: 1
Tree-based Models

- **Empirical successes**
  - Information retrieval [LambdaMART; Burges, 2010]
  - Computer vision [Babenko et al., 2011]
  - Real world classification [Fernandez-Delgado et al., 2014]

- **Why tree-based models?**
  - Handle categorical features and count data better.
  - Implicitly perform feature selection.
We present **S-MART**: **Structured Multiple Additive Regression Trees**

- A general class of tree-based structured learning algorithms.
- A friend of problems with dense features.

We apply S-MART to entity linking on short and noisy texts

- Entity linking utilizes statistics dense features.

Experimental results show that S-MART significantly outperforms all alternative baselines.
Outline

- S-MART: A family of Tree-based Structured Learning Algorithms
- S-MART for Tweet Entity Linking
  - Non-overlapping inference
- Experiments
Outline

- S-MART: A family of Tree-based Structured Learning Algorithms
- S-MART for Tweet Entity Linking
  - Non-overlapping inference
- Experiments
Structured Learning

- Model a joint scoring function $S(x, y)$ over an input structure $x$ and an output structure $y$

- Obtain the prediction requires inference (e.g., dynamic programming)

$$\hat{y} = \arg \max_{y \in \text{Gen}(x)} S(x, y)$$
Structured Multiple Additive Regression Trees (S-MART)

- Assume a decomposition over factors
  \[ S(x, y) = \sum_{k \in \Omega(x)} F(x, y_k) \]

- Optimize with functional gradient descents
  \[ F_m(x, y_k) = F_{m-1}(x, y_k) - \eta_m g_m(x, y_k) \]

- Model functional gradients using regression trees \( h_m(x, y_k) \)
  \[ F(x, y_k) = F_M(x, y_k) = \sum_{m=1}^{M} \eta_m h_m(x, y_k) \]
Gradient Descent

- Linear combination of parameters and feature functions

$$F(x, y_k) = \mathbf{w}^\top f(x, y_k)$$

- Gradient descent in vector space

$$\mathbf{w}_m = \mathbf{w}_{m-1} - \eta_m \frac{\partial L}{\partial \mathbf{w}_{m-1}}$$
Gradient Descent in Function Space

\[ F_0(x, y_k) = 0 \]

\[ F_{m-1}(x, y_k) \]

\[ g_m(x, y_k) = \left[ \frac{\partial L(y^*, S(x, y_k))}{\partial F(x, y_k)} \right] \]

\[ F(x, y_k) = F_{m-1}(x, y_k) \]
Gradient Descent in Function Space

\[ F_0(x, y_k) = 0 \]

Requiring Inference

\[ g_m(x, y_k) = \left[ \frac{\partial L(y^*, S(x, y_k))}{\partial F(x, y_k)} \right] \]

\[ F(x, y_k) = F_{m-1}(x, y_k) \]
Gradient Descent in Function Space

\[ F_0(x, y_k) = 0 \]

\[ F_m(x, y_k) = F_{m-1}(x, y_k) - \eta_m g_m(x, y_k) \]
Model Functional Gradients

\[-g_m(x, y_k)\]
Model Functional Gradients

- Pointwise Functional Gradients
Model Functional Gradients

- Pointwise Functional Gradients
  - Approximation by regression

\[ -g_m(x, y_k) \]

Tree

- Is linkprob > 0.5?
  - Is PER?
    - \(-0.5\)
  - \(-0.1\)
    - Is clickprob > 0.1?
      - \(-0.3\)
      - \(\ldots\)
S-MART vs. TreeCRF

Structure
- Linear chain
- Various structures

Loss function
- Logistic loss
- Various losses

Scoring function

[Dietterich +, 2004]
S-MART vs. TreeCRF

TreeCRF
[Dietterich+, 2004]

S-MART
S-MART vs. TreeCRF

TreeCRF
[Dietterich+, 2004]

S-MART

Structure

Linear chain

Various structures
## S-MART vs. TreeCRF

<table>
<thead>
<tr>
<th></th>
<th>TreeCRF [Dietterich+, 2004]</th>
<th>S-MART</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structure</strong></td>
<td>Linear chain</td>
<td>Various structures</td>
</tr>
<tr>
<td><strong>Loss function</strong></td>
<td>Logistic loss</td>
<td>Various losses</td>
</tr>
</tbody>
</table>
## S-MART vs. TreeCRF

<table>
<thead>
<tr>
<th>Structure</th>
<th>TreeCRF [Dietterich+, 2004]</th>
<th>S-MART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear chain</td>
<td>Various structures</td>
<td></td>
</tr>
</tbody>
</table>

| Loss function   | Logistic loss               | Various losses |

| Scoring function | $F^{y_t}(x)$               | $F(x, y_t)$ |

TreeCRF and S-MART are compared based on their structure, loss function, and scoring function. S-MART supports various structures and losses, while TreeCRF uses a linear chain structure and logistic loss.
Outline

‣ S-MART: A family of Tree-based Structured Learning Algorithms

‣ S-MART for Tweet Entity Linking
  ▸ Non-overlapping inference

‣ Experiments
Entity Linking in Short Texts

- Data explosion: noisy and short texts
  - Twitter messages
  - Web queries

- Downstream applications
  - Semantic parsing and question answering [Yih et al., 2015]
  - Relation extraction [Riedel et al., 2013]
Tweet Entity Linking

@TaylorYanda
Eli Manning and the New York Giants are going to win the World Series

#Game7
Entity Linking meets Dense Features

- Short of labeled data
  - Lack of context makes annotation more challenging.
  - Language changes, annotation may become stale and ill-suited for new spellings and words. [Yang and Eisenstein, 2013]

- Powerful statistic dense features [Guo et al., 2013]
  - The probability of a surface form to be an entity
  - View count of a Wikipedia page
  - Textual similarity between a tweet and a Wikipedia page
Structured learning: select the best non-overlapping entity assignment
- Choose top 20 entity candidates for each surface form
- Add a special NIL entity to represent no entity should be fired here

Eli Manning and the New York Giants are going to win the World Series
Structured learning: select the best non-overlapping entity assignment
- Choose top 20 entity candidates for each surface form
- Add a special NIL entity to represent no entity should be fired here

Eli Manning and the New York Giants are going to win the World Series
Structured learning: select the best non-overlapping entity assignment
- Choose top 20 entity candidates for each surface form
- Add a special NIL entity to represent no entity should be fired here

Eli Manning and the New York Giants are going to win the World Series
Structured learning: select the best non-overlapping entity assignment
  - Choose top 20 entity candidates for each surface form
  - Add a special NIL entity to represent no entity should be fired here

Eli Manning and the New York Giants are going to win the World Series

Eli_Manning
- **Structured learning**: select the best non-overlapping entity assignment
  - Choose top 20 entity candidates for each surface form
  - Add a special NIL entity to represent no entity should be fired here

*Eli Manning and the New York Giants are going to win the World Series*
S-MART for Tweet Entity Linking

- **Logistic loss**

\[
L(y^*, S(x, y)) = - \log P(y^* | x)
\]

\[
= \log Z(x) - S(x, y^*)
\]

- **Point-wise gradients**

\[
g_{ku} = \frac{\partial L}{\partial F(x, y_k = u_k)}
\]

\[
= P(y_k = u_k | x) - 1[y_k^* = u_k]
\]
Eli Manning and the New York Giants are going to win the World Series.
Eli Manning and the New York Giants are going to win the World Series.

\[ \beta(u_k, k) \]
Eli Manning and the New York Giants are going to win the World Series.
Outline

- S-MART: A family of Tree-based Structured Learning Algorithms
- S-MART for Tweet Entity Linking
  - Non-overlapping inference
- Experiments
### Data

- **Named Entity Extraction & Linking (NEEL) Challenge datasets** [Cano et al., 2014]
- **TACL datasets** [Fang & Chang, 2014]

<table>
<thead>
<tr>
<th>Data</th>
<th>#Tweet</th>
<th>#Entity</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEEL Train</td>
<td>2,340</td>
<td>2,202</td>
<td>Jul. ~ Aug. 11</td>
</tr>
<tr>
<td>NEEL Test</td>
<td>1,164</td>
<td>687</td>
<td>Jul. ~ Aug. 11</td>
</tr>
<tr>
<td>TACL-IE</td>
<td>500</td>
<td>300</td>
<td>Dec. 12</td>
</tr>
<tr>
<td>TACL-IR</td>
<td>980</td>
<td>-</td>
<td>Dec. 12</td>
</tr>
</tbody>
</table>
Evaluation Methodology

- **IE-driven Evaluation** [Guo et al., 2013]
  - Standard evaluation of the system ability on extracting entities from tweets
  - Metric: macro F-score

- **IR-driven Evaluation** [Fang & Chang, 2014]
  - Evaluation of the system ability on disambiguation of the target entities in tweets
  - Metric: macro F-score on query entities
### Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Structured</th>
<th>Non-linear</th>
<th>Tree-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured Perceptron</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear SSVM*</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Polynomial SSVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LambdaRank</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>MART#</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>S-MART</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* previous state of the art system
# winning system of NEEL challenge 2014
# IE-driven Evaluation

<table>
<thead>
<tr>
<th>NEEL Test F1</th>
<th>Linear SSVM</th>
<th>Poly SSVM</th>
<th>LambdaRank</th>
<th>MART</th>
<th>S-MART</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IE-driven Evaluation

<table>
<thead>
<tr>
<th>SP</th>
<th>Linear SSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>70.9</td>
<td>73.2</td>
</tr>
</tbody>
</table>

Linear SSVM outperforms SP, achieving a higher F1 score.
IE-driven Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>Linear</th>
<th>Non-linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SSVM</td>
<td>73.2</td>
<td>74.6</td>
</tr>
<tr>
<td>Poly SSVM</td>
<td>70.9</td>
<td>75.5</td>
</tr>
<tr>
<td>LambdaRank</td>
<td>71.4</td>
<td>75.5</td>
</tr>
</tbody>
</table>

The chart compares different methods (SP, Linear SSVM, Poly SSVM, LambdaRank) for linear and non-linear models, with F1 scores in the range of 65 to 85.
IE-driven Evaluation

![Graph showing NEEL Test F1 scores for different models: SP, Linear SSVM, Poly SSVM, LambdaRank. The graph highlights the performance of a Neural based model with scores of 70.9, 73.2, 74.6, and 75.5.]
## IE-driven Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>NEEL Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>70.9</td>
</tr>
<tr>
<td>Linear SSVM</td>
<td>73.2</td>
</tr>
<tr>
<td>Poly SSVM</td>
<td>74.6</td>
</tr>
<tr>
<td>LambdaRank</td>
<td>75.5</td>
</tr>
</tbody>
</table>

**Kernel based model**

![Bar chart](chart.png)
IE-driven Evaluation

Tree based model

Linear

70.9 73.2 74.6 75.5 77.4
IE-driven Evaluation

- SP
- Linear SSVM
- Poly SSVM
- LambdaRank
- MART
- S-MART

Tree based structured model

<table>
<thead>
<tr>
<th>Model</th>
<th>NEEL Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>70.9</td>
</tr>
<tr>
<td>Linear SSVM</td>
<td>73.2</td>
</tr>
<tr>
<td>Poly SSVM</td>
<td>74.6</td>
</tr>
<tr>
<td>LambdaRank</td>
<td>75.5</td>
</tr>
<tr>
<td>MART</td>
<td>77.4</td>
</tr>
<tr>
<td>S-MART</td>
<td>81.1</td>
</tr>
</tbody>
</table>
IR-driven Evaluation

<table>
<thead>
<tr>
<th>TACL-IR F1</th>
<th>Linear SSVM</th>
<th>Poly SSVM</th>
<th>LambdaRank</th>
<th>MART</th>
<th>S-MART</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IR-driven Evaluation

SP | Linear SSVM
--- | ---
58.0 | 62.2
IR-driven Evaluation

<table>
<thead>
<tr>
<th></th>
<th>SP</th>
<th>Linear SSVM</th>
<th>Poly SSVM</th>
<th>LambdaRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>TACL-IR F1</td>
<td></td>
<td>58.0</td>
<td>62.2</td>
<td>56.8</td>
</tr>
</tbody>
</table>
IR-driven Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>TACL-IR F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>58.0</td>
</tr>
<tr>
<td>Linear SSVM</td>
<td>62.2</td>
</tr>
<tr>
<td>Poly SSVM</td>
<td>63.6</td>
</tr>
<tr>
<td>LambdaRank</td>
<td>56.8</td>
</tr>
<tr>
<td>MART</td>
<td>63.0</td>
</tr>
</tbody>
</table>
IR-driven Evaluation

TACL-IR F1

<table>
<thead>
<tr>
<th>Method</th>
<th>Linear</th>
<th>Non-linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>58.0</td>
<td>67.4</td>
</tr>
<tr>
<td>Linear SSVM</td>
<td>62.2</td>
<td>63.6</td>
</tr>
<tr>
<td>Poly SSVM</td>
<td>63.6</td>
<td>56.8</td>
</tr>
<tr>
<td>LambdaRank</td>
<td>56.8</td>
<td></td>
</tr>
<tr>
<td>MART</td>
<td>63.0</td>
<td></td>
</tr>
<tr>
<td>S-MART</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Linear
- Non-linear
Conclusion

- A novel tree-based structured learning framework S-MART
  - Generalization of TreeCRF
- A novel inference algorithm for non-overlapping structure of the tweet entity linking task.
- **Application**: Knowledge base QA (outstanding paper of ACL’15)
  - Our system is a core component of the QA system.
- Rise of non-linear models
  - We can try advanced neural based structured algorithms
  - It’s worth to try different non-linear models
Thank you!