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
Traditional NLP Settings

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

- Low dimensional embedding features
Rise of Dense Features

- Low dimensional embedding features
- Low dimensional statistics features
- Named mention statistics
- Click-through statistics
Rise of Dense Features

- Low dimensional embedding features
- Low dimensional statistics features

Named mention statistics
Click-through statistics

Dense features + Non-linear models
Non-linear Models
Non-linear Models

- Neural networks
Non-linear Models

- Neural networks

![Bar chart showing # of Papers in ACL'15](chart.png)
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)
Tree-based Models
Tree-based Models

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

- 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 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.
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 $y$
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)
Structured Multiple Additive Regression Trees (S-MART)

- Assume a decomposition over factors

\[ S(x, y) = \sum_{k \in \Omega(x)} F(x, y_k) \]
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
  \[ F_m(x, y_k) = F_{m-1}(x, y_k) - \eta_m g_m(x, y_k) \]
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
Gradient Descent

- Linear combination of parameters and feature functions

\[ F(x, y_k) = w^\top f(x, y_k) \]
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

- Linear combination of parameters and feature functions

\[ F(x, y_k) = w^T f(x, y_k) \]

- Gradient descent in vector space

\[ w_m = w_{m-1} - \eta_m \frac{\partial L}{\partial w_{m-1}} \]
Gradient Descent in Function Space
Gradient Descent in Function Space

\[ F_0(x, y_k) = 0 \]
Gradient Descent in Function Space

\[ F_0(x, y_k) = 0 \]
Gradient Descent in Function Space

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

\[ g_m(x, y_k) = \left[ \frac{\partial L(y^*, S(x, y_k))}{\partial F(x, y_k)} \right] \quad F(x, y_k) = F_{m-1}(x, y_k) \]
Gradient Descent in Function Space

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

\[ g_m(x, y_k) = \left[ \frac{\partial L(y^*, S(x, y_k))}{\partial F(x, y_k)} \right] F(x, y_k) = F_{m-1}(x, y_k) \]
Gradient Descent in Function Space

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

\[ g_m(x, y_k) = \left[ \frac{\partial L(y^*, S(x, y_k))}{\partial F(x, y_k)} \right] F(x, y_k) = F_{m-1}(x, y_k) \]

Requiring Inference
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

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

- Pointwise Functional Gradients
Model Functional Gradients

- Pointwise Functional Gradients
  - Approximation by regression

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

- Pointwise Functional Gradients
  - Approximation by regression

![Graph showing functional gradient with points and tree structure]

\[-g_m(x, y_k)\]

- Is linkprob > 0.5?
  - Is PER?
    - -0.5
    - -0.1
    - Is clickprob > 0.1?
      - -0.3
      - ...
S-MART vs. TreeCRF

Structure: Linear chain vs. Various structures
Loss function: Logistic loss vs. Various losses
Scoring function
S-MART vs. TreeCRF

TreeCRF
[Dietterich+, 2004]

S-MART

- **Structure**: Linear chain vs. Various structures
- **Loss function**: Logistic loss vs. Various losses
- **Scoring function**
## 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></td>
<td>Various structures</td>
</tr>
</tbody>
</table>
## 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></th>
<th>TreeCRF</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>
<tr>
<td><strong>Scoring function</strong></td>
<td>$F^{y_t}(x)$</td>
<td>$F(x, y_t)$</td>
</tr>
</tbody>
</table>

[Dietterich+, 2004]
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
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]
Eli Manning and the New York Giants are going to win the World Series.

Game7
Tweet Entity Linking

Yanda @TaylorYanda · 33s

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

#Game7
Tweet Entity Linking

Eli Manning and the New York Giants are going to win the World Series.
Tweet Entity Linking

Yanda @TaylorYanda · 33s

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

#Game7
Entity Linking meets Dense Features
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]
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
System Overview

Tokenized Message → Candidate Generation → Joint Recognition and Disambiguation → Entity Linking Results
- **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
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
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
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*
System Overview

- **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

Tokenized Message → Candidate Generation → Joint Recognition and Disambiguation → Entity Linking Results

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

- **Eli Manning**
- **New York Giants**
- **World Series**
S-MART for Tweet Entity Linking
Logistic loss

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

\[
g_{ku} = \frac{\partial L}{\partial F(x, y_k = u_k)} = P(y_k = u_k | x) - 1[y_k^* = u_k]
\]
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.
Eli Manning and the New York Giants are going to win the World Series.

\[
\alpha(u_k, k) = \exp\left( F(x, y_k = u_k) \right) \\
\prod_{p=1}^{P-1} \exp\left( F(x, y_{k-p} = \text{Nil}) \right) \\
\sum_{u_{k-P}} \alpha(u_{k-P}, k - P)
\]
Eli Manning and the New York Giants are going to win the World Series.

\[
\alpha(u_k, k) = \exp\left(F(x, y_k = u_k)\right) \\
\prod_{p=1}^{P-1} \exp(F(x, y_{k-p} = \text{Nil})) \\
\sum_{u_{k-P}} \alpha(u_{k-P}, k-P)
\]
Inference: Forward Algorithm

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

\[ \alpha(u_k, k) = \exp \left( F \left( x, y_k = u_k \right) \right) \]

\[ \cdot \prod_{p=1}^{P-1} \exp \left( F \left( x, y_{k-p} = \text{Nil} \right) \right) \]

\[ \cdot \sum_{u_{k-P}} \alpha(u_{k-P}, k - P) \]
Eli Manning and the New York Giants are going to win the World Series.

\[
\alpha(u_k, k) = \exp\left(F(x, y_k = u_k)\right)
\prod_{p=1}^{P-1} \exp\left(F(x, y_{k-p} = \text{Nil})\right)
\sum_{u_{k-P}} \alpha(u_{k-P}, k - P)
\]
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.

\[ \beta(u_k, k) \]
Inference: Backward Algorithm

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

\[ \beta(u_k, k) = \sum_{u_{k+Q}} \exp(F(x, y_{k+Q} = u_{k+Q})) \cdot \prod_{q=1}^{Q-1} \exp(F(x, y_{k+q} = \text{Nil})) \cdot \beta(u_{k+Q}, k + Q) \]
Inference: Backward Algorithm

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.
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]
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
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>0.763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
<td>0.665</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td></td>
<td></td>
<td>0.732</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td></td>
<td></td>
<td>0.746</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td></td>
<td></td>
<td>0.755</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IE-driven Evaluation

- SP
- Linear SSVM

<table>
<thead>
<tr>
<th>Method</th>
<th>NEEL Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SSVM</td>
<td>73.2</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>78.4</td>
</tr>
</tbody>
</table>

Scores: 70.9, 73.2
IE-driven Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>NEEL Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SSVM</td>
<td>73.2</td>
</tr>
<tr>
<td>Linear</td>
<td>73.2</td>
</tr>
<tr>
<td>LambdaRank</td>
<td></td>
</tr>
<tr>
<td>MART</td>
<td></td>
</tr>
<tr>
<td>S-MART</td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- SP
- Linear SSVM
IE-driven Evaluation

SP | Linear SSVM | Poly SSVM | LambdaRank
---|-------------|-----------|-------------
70.9 | 73.2 | 74.6 | 75.5

Bar chart showing NEEL Test F1 scores for Linear, Linear SSVM, Poly SSVM, and LambdaRank models.
IE-driven Evaluation

- SP
- Linear SSVM
- Poly SSVM
- LambdaRank

<table>
<thead>
<tr>
<th>Method</th>
<th>Linear</th>
<th>Non-linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>70.9</td>
<td>75.5</td>
</tr>
<tr>
<td>Linear SSVM</td>
<td>73.2</td>
<td>74.6</td>
</tr>
<tr>
<td>LambdaRank</td>
<td>85.9</td>
<td>85.9</td>
</tr>
<tr>
<td>Poly SSVM</td>
<td>85.9</td>
<td>85.9</td>
</tr>
<tr>
<td>MART</td>
<td>75.9</td>
<td>75.9</td>
</tr>
<tr>
<td>S-MART</td>
<td>75.9</td>
<td>75.9</td>
</tr>
</tbody>
</table>
IE-driven Evaluation

<table>
<thead>
<tr>
<th>Model</th>
<th>NEEL Test F1</th>
</tr>
</thead>
<tbody>
<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>

- SP
- Linear SSVM
- Poly SSVM
- LambdaRank
IE-driven Evaluation

- Linear SSVM
- Poly SSVM
- LambdaRank

<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</td>
<td>73.2</td>
</tr>
<tr>
<td>Linear</td>
<td>74.6</td>
</tr>
<tr>
<td>Linear</td>
<td>75.5</td>
</tr>
</tbody>
</table>

Neural based model
IE-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>Linear</td>
<td>70.9</td>
<td>73.2</td>
<td>74.6</td>
<td>75.5</td>
</tr>
</tbody>
</table>
IE-driven Evaluation

- SP
- Linear SSVM
- Poly SSVM
- LambdaRank

Kernel based model

- Linear: 70.9
- Linear SSVM: 73.2
- Poly SSVM: 74.6
- LambdaRank: 75.5

NEEL Test F1
IE-driven Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>Linear SSVM</th>
<th>Poly SSVM</th>
<th>LambdaRank</th>
<th>MART</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEEL Test F1</td>
<td>73.2</td>
<td>74.6</td>
<td>75.5</td>
<td>77.4</td>
</tr>
</tbody>
</table>

- Linear SSVM: 73.2
- Poly SSVM: 74.6
- LambdaRank: 75.5
- MART: 77.4
IE-driven Evaluation

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

Tree based model

<table>
<thead>
<tr>
<th>Model</th>
<th>NEEL Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>70.9</td>
</tr>
<tr>
<td>Linear</td>
<td>73.2</td>
</tr>
<tr>
<td>Linear</td>
<td>74.6</td>
</tr>
<tr>
<td>Linear</td>
<td>75.5</td>
</tr>
<tr>
<td>Linear</td>
<td>77.4</td>
</tr>
</tbody>
</table>
IE-driven Evaluation

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

<table>
<thead>
<tr>
<th>Method</th>
<th>65</th>
<th>70</th>
<th>75</th>
<th>80</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SSVM</td>
<td>70.9</td>
<td>73.2</td>
<td>74.6</td>
<td>75.5</td>
<td>81.1</td>
</tr>
<tr>
<td>Poly SSVM</td>
<td>70.9</td>
<td>73.2</td>
<td>74.6</td>
<td>75.5</td>
<td>81.1</td>
</tr>
<tr>
<td>LambdaRank</td>
<td>70.9</td>
<td>73.2</td>
<td>74.6</td>
<td>75.5</td>
<td>81.1</td>
</tr>
<tr>
<td>MART</td>
<td>70.9</td>
<td>73.2</td>
<td>74.6</td>
<td>75.5</td>
<td>81.1</td>
</tr>
<tr>
<td>S-MART</td>
<td>70.9</td>
<td>73.2</td>
<td>74.6</td>
<td>75.5</td>
<td>81.1</td>
</tr>
</tbody>
</table>
IE-driven Evaluation

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

Tree based structured model

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

### TACL-IR F1

<table>
<thead>
<tr>
<th>Method</th>
<th>SP</th>
<th>Linear SSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>58.0</td>
<td>62.2</td>
</tr>
</tbody>
</table>

### Diagram

- **SP**
- **Linear SSVM**
IR-driven Evaluation

<table>
<thead>
<tr>
<th>Method</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>58.0</td>
<td>62.2</td>
<td>63.6</td>
<td>56.8</td>
</tr>
</tbody>
</table>
IR-driven Evaluation

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

<table>
<thead>
<tr>
<th>Method</th>
<th>SP</th>
<th>Linear SSVM</th>
<th>Poly SSVM</th>
<th>LambdaRank</th>
<th>MART</th>
<th>S-MART</th>
</tr>
</thead>
<tbody>
<tr>
<td>TACL-IR F1</td>
<td>58.0</td>
<td>62.2</td>
<td>63.6</td>
<td>56.8</td>
<td>63.0</td>
<td>67.4</td>
</tr>
</tbody>
</table>
IR-driven Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>TACL-IR F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</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>
<tr>
<td>S-MART</td>
<td>67.4</td>
</tr>
</tbody>
</table>

Bar chart showing TACL-IR F1 scores for different methods: SP, Linear SSVM, Poly SSVM, LambdaRank, MART, S-MART.
IR-driven Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>F1 Score Linear</th>
<th>F1 Score 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>

Bar chart showing TACL-IR F1 scores for different methods and models.
Conclusion
A novel tree-based structured learning framework S-MART

Generalization of TreeCRF
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.
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.
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!