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

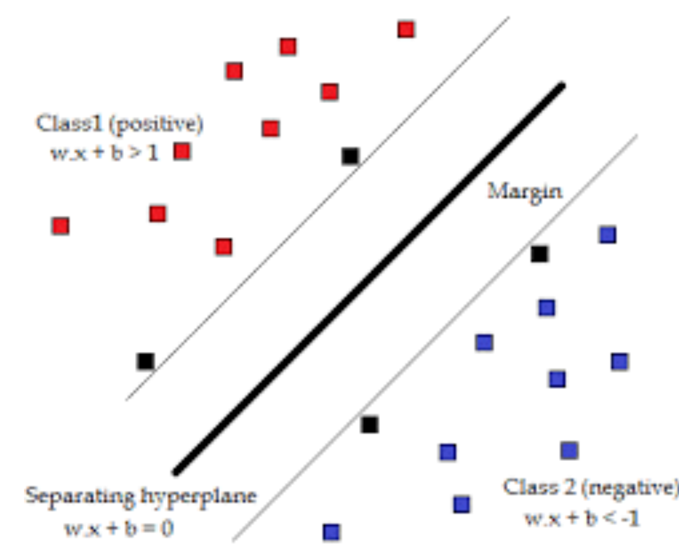
Yi Yang* and Ming-Wei Chang#

*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

Named mention statistics
Click-through statistics

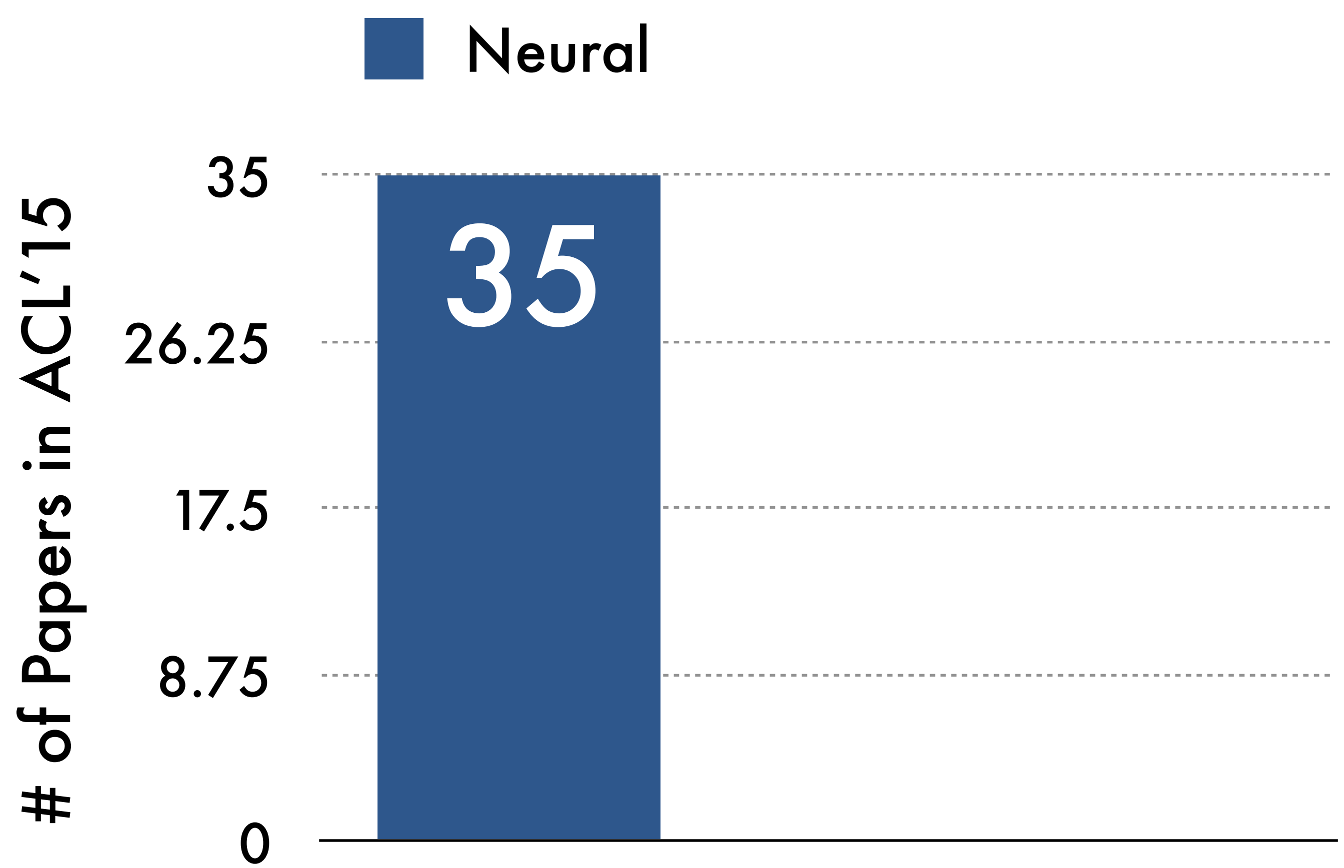
Dense features + Non-linear models

Non-linear Models

- Neural networks

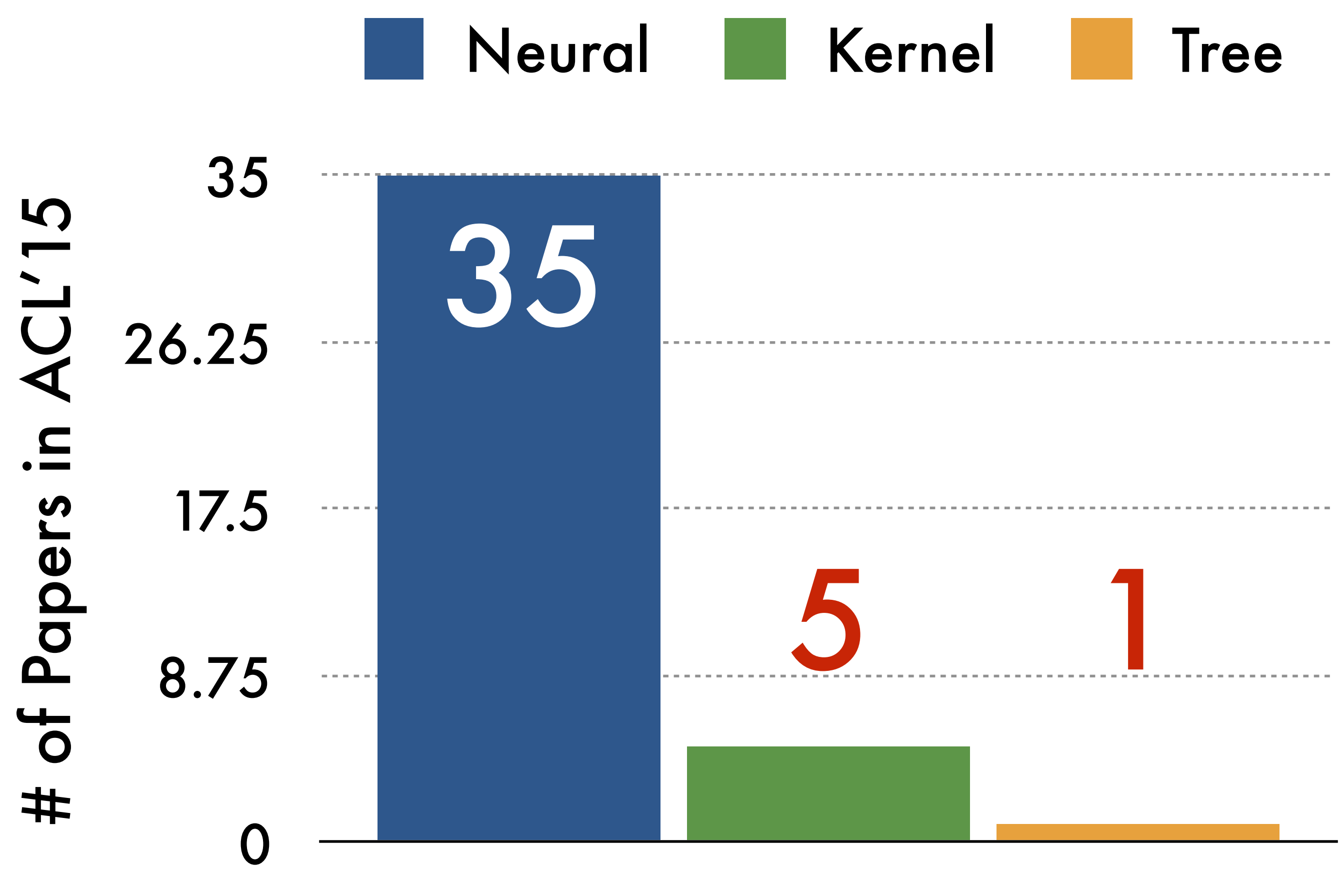
Non-linear Models

- 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)



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 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 $S(\mathrm{x}, \mathrm{y})$ over an input structure x and an output structure y

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

$$\widehat{\mathbf{y}} = \arg\max_{y \in Gen(\mathbf{x})} S(\mathbf{x}, \mathbf{y})$$

Structured Multiple Additive Regression Trees (S-MART)

- Assume a decomposition over factors

$$S(\mathbf{x}, \mathbf{y}) = \sum_{k \in \Omega(\mathbf{x})} F(\mathbf{x}, \mathbf{y}_k)$$

- Optimize with functional gradient descents

$$F_m(\mathbf{x}, \mathbf{y}_k) = F_{m-1}(\mathbf{x}, \mathbf{y}_k) - \eta_m g_m(\mathbf{x}, \mathbf{y}_k)$$

- Model functional gradients using regression trees $h_m(\mathbf{x}, \mathbf{y}_k)$

$$F(\mathbf{x}, \mathbf{y}_k) = F_M(\mathbf{x}, \mathbf{y}_k) = \sum_{m=1}^{M} \eta_m h_m(\mathbf{x}, \mathbf{y}_k)$$

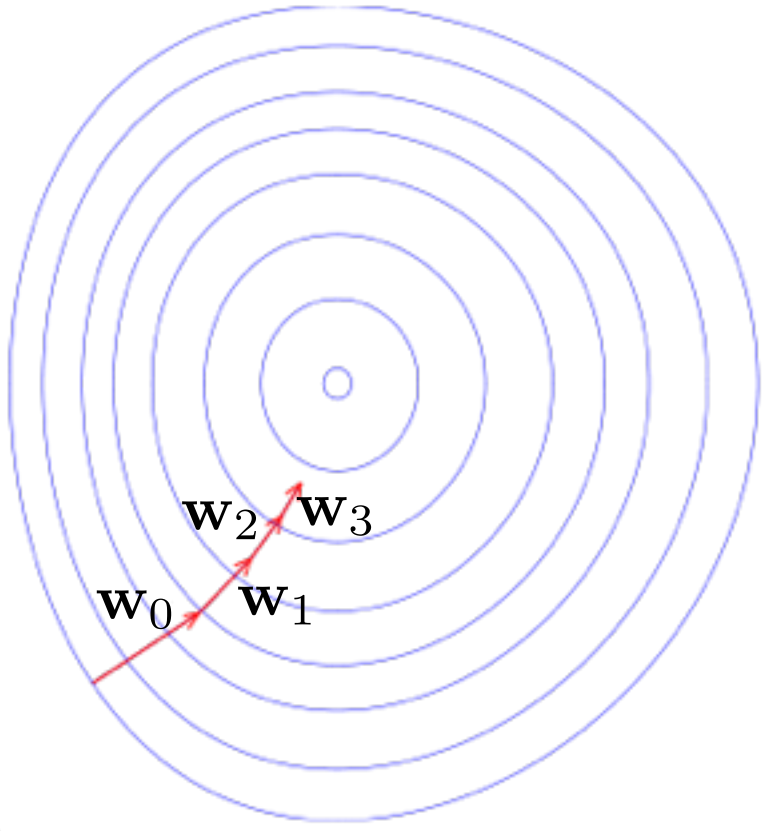
Gradient Descent

- Linear combination of parameters and feature functions

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

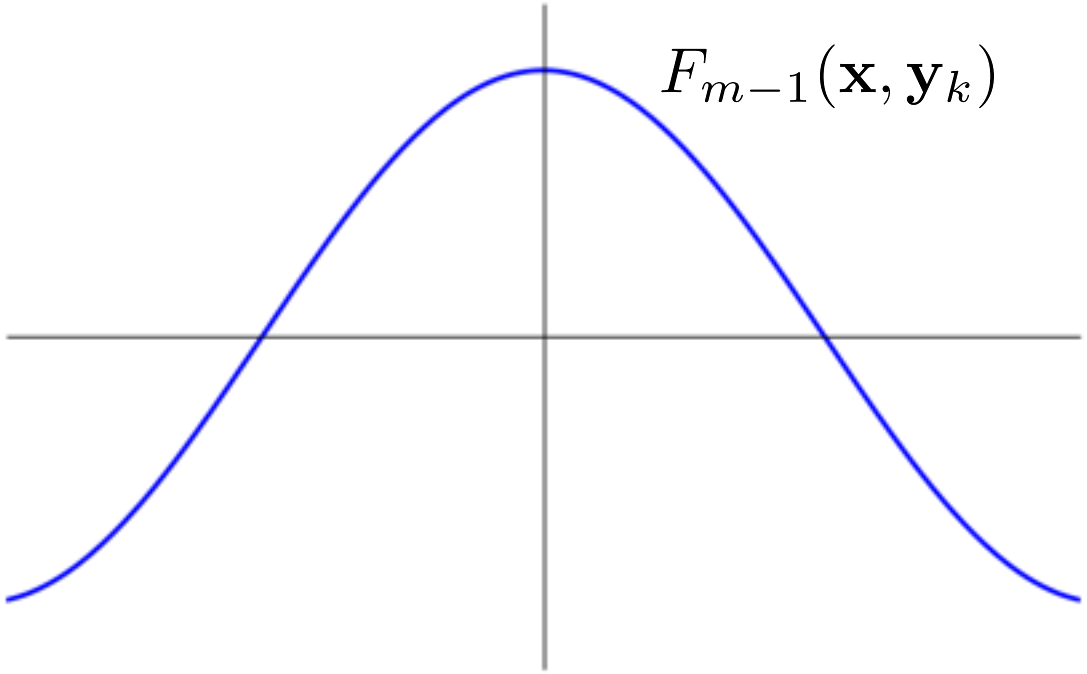
- Gradient descent in vector space

$$\mathbf{w}_m = \mathbf{w}_{m-1} - \eta_m \frac{\partial L}{\partial \mathbf{w}_{m-1}}$$



Gradient Descent in Function Space

$$F_0(\mathbf{x}, \mathbf{y}_k) = 0$$



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

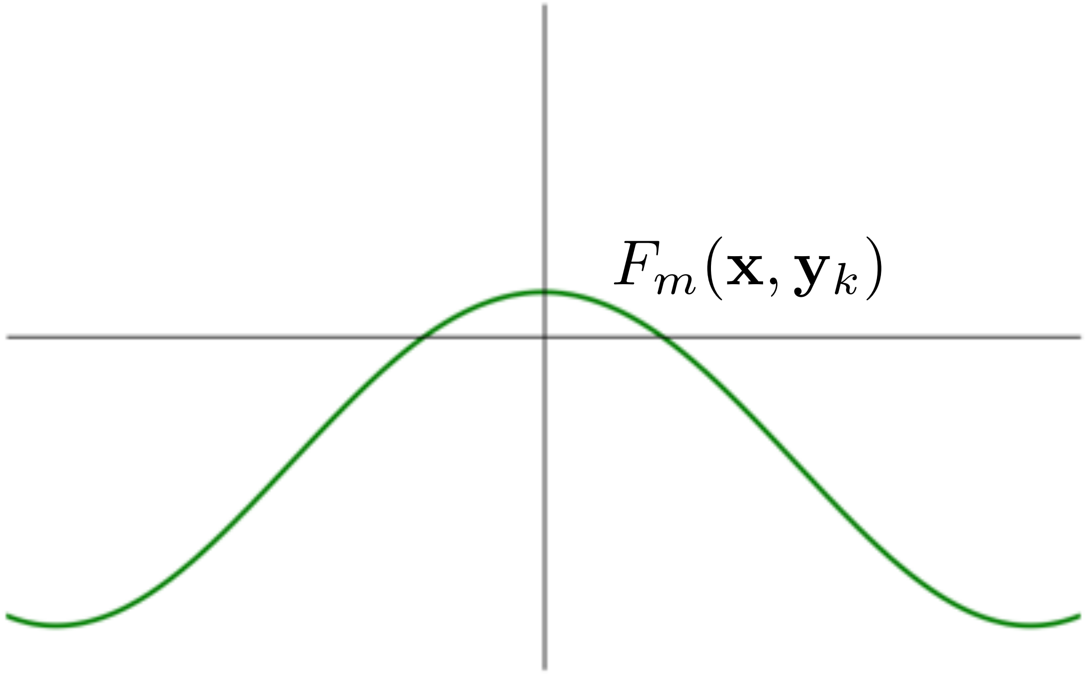
Gradient Descent in Function Space

$$F_0(\mathbf{x}, \mathbf{y}_k) = 0$$

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

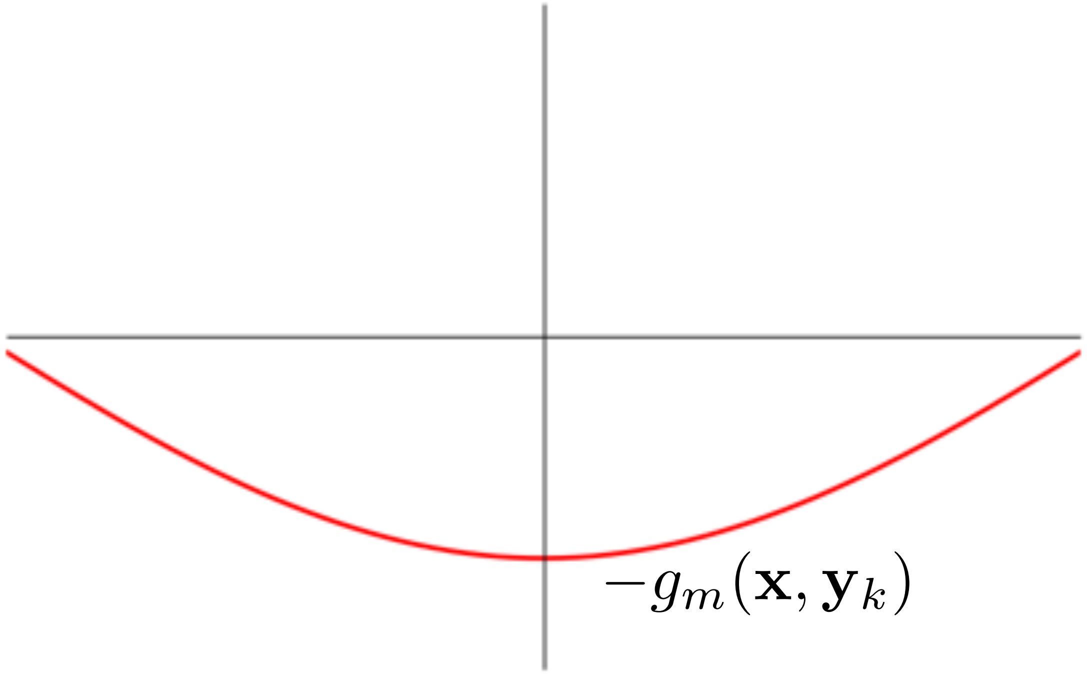
Gradient Descent in Function Space

$$F_0(\mathbf{x}, \mathbf{y}_k) = 0$$



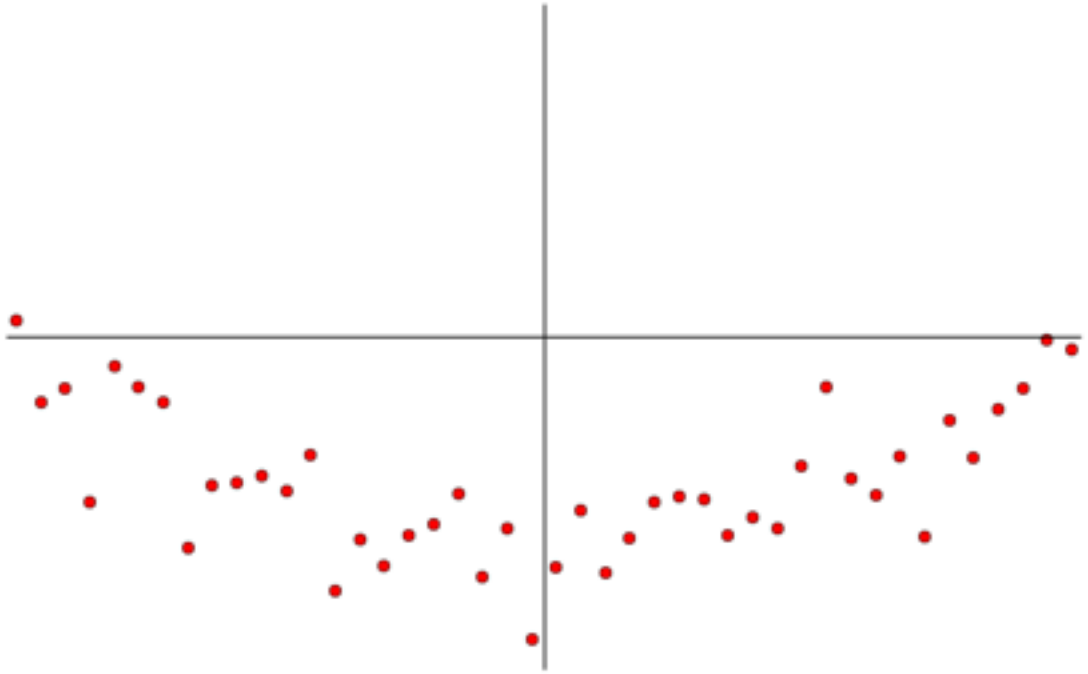
$$F_m(\mathbf{x}, \mathbf{y}_k) = F_{m-1}(\mathbf{x}, \mathbf{y}_k) - \eta_m g_m(\mathbf{x}, \mathbf{y}_k)$$

Model Functional Gradients



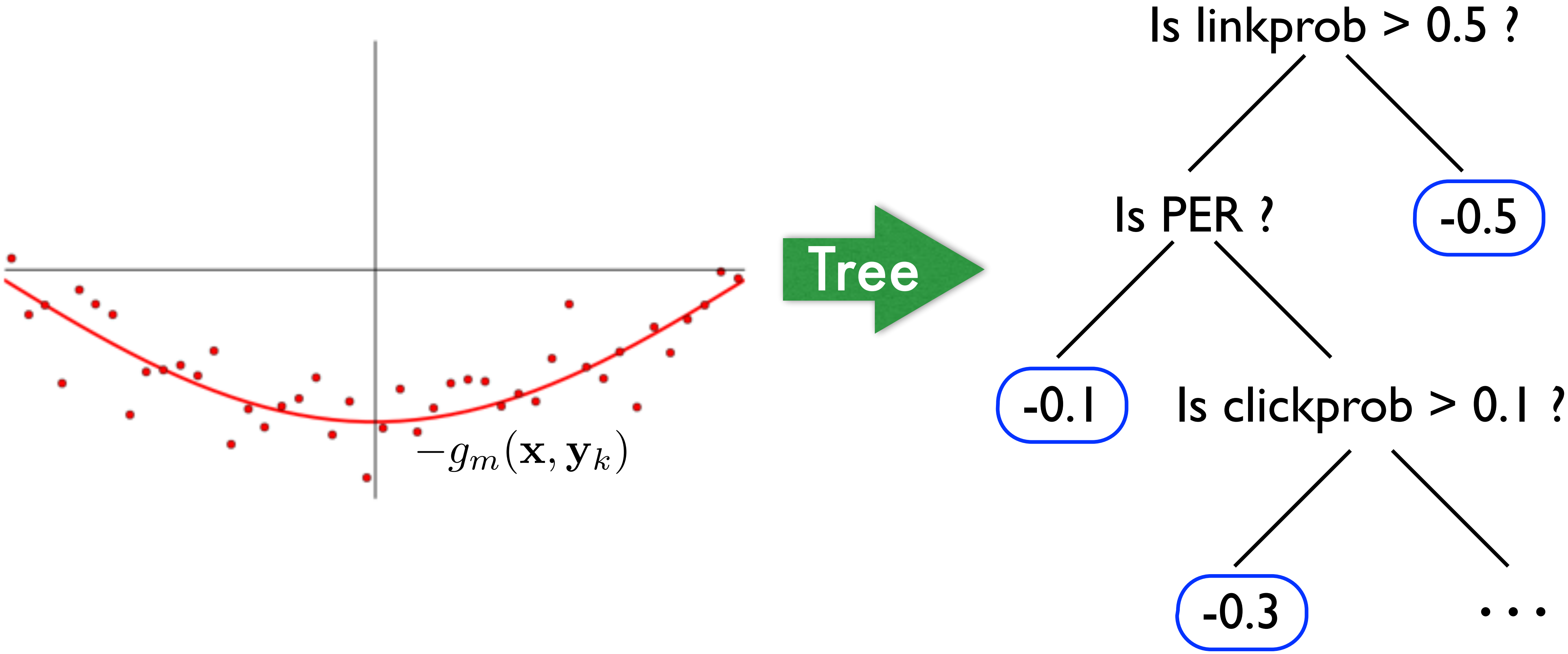
Model Functional Gradients

- Pointwise Functional Gradients



Model Functional Gradients

- Pointwise Functional Gradients
  - Approximation by regression



S-MART vs. TreeCRF

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

	TreeCRF [Dietterich+, 2004]	S-MART
Structure	Linear chain	Various structures
Loss function	Logistic loss	Various losses

S-MART vs. TreeCRF

	TreeCRF [Dietterich+, 2004]	S-MART
Structure	Linear chain	Various structures
Loss function	Logistic loss	Various losses
Scoring function	$F^{y_t}(\mathbf{x})$	$F(\mathbf{x}, \mathbf{y}_t)$

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]

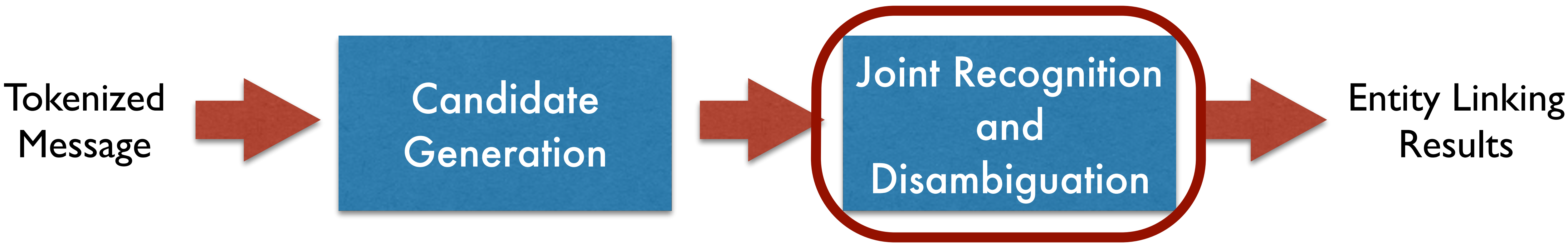
Tweet Entity Linking

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

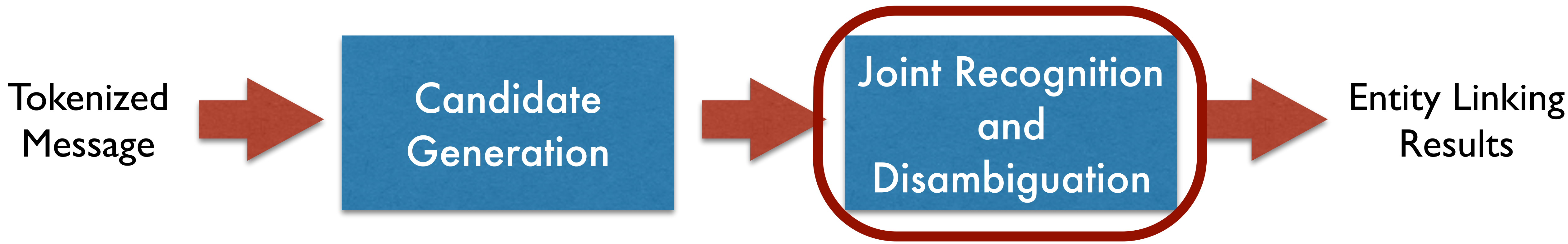
System Overview



- 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

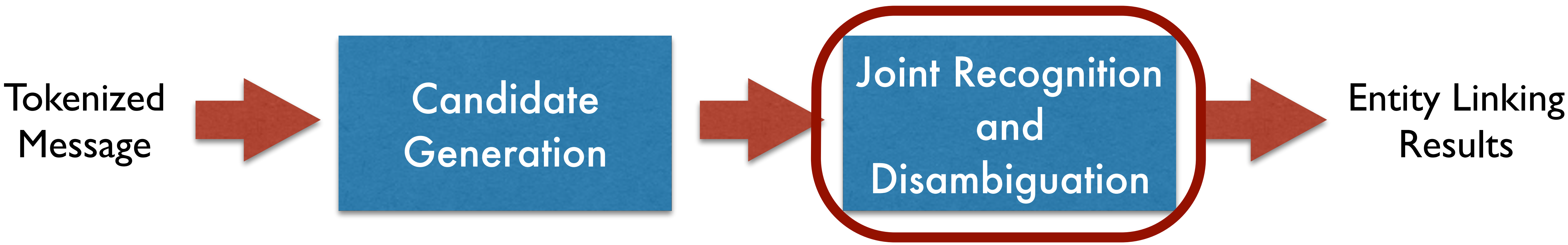


- Structured learning: select the best non-overlapping entity assignment
  - Choose top 20 entity candidates for each surface form
  - Add a special NIL entity to represent no entity should be fired here

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

NIL

System Overview

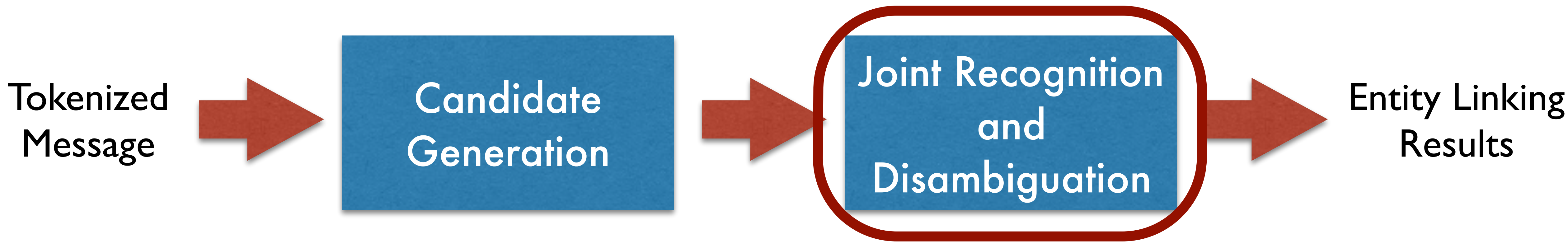


- Structured learning: select the best non-overlapping entity assignment
  - Choose top 20 entity candidates for each surface form
  - Add a special NIL entity to represent no entity should be fired here

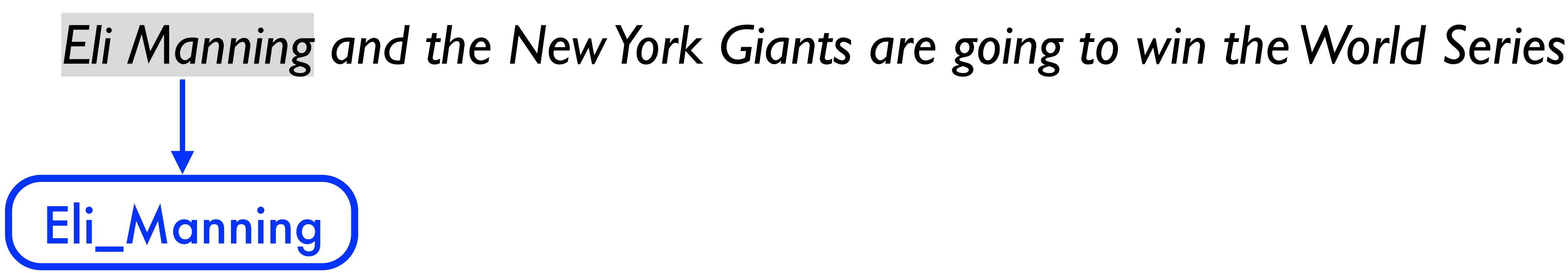
Eli Manning and the New York Giants are going to win the World Series

NIL

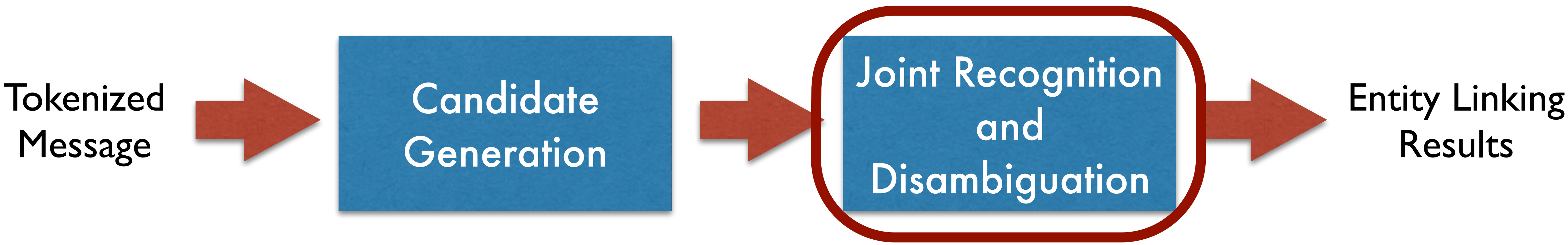
System Overview



- 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



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

S-MART for Tweet Entity Linking

- Logistic loss

$$\begin{aligned} L(\mathbf{y}^*, S(\mathbf{x}, \mathbf{y})) &= -\log P(\mathbf{y}^* | \mathbf{x}) \\ &= \log Z(\mathbf{x}) - S(\mathbf{x}, \mathbf{y}^*) \end{aligned}$$

- Point-wise gradients

Non-overlapping Inference

$$\begin{aligned} g_{ku} &= \frac{\partial L}{\partial F(\mathbf{x}, y_k = u_k)} \\ &= \boxed{P(y_k = u_k | \mathbf{x})} - \mathbf{1}[y_k^* = u_k] \end{aligned}$$

Inference: Forward Algorithm

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

$$\alpha(u_k, k) = \exp(F(\mathbf{x}, y_k = u_k)) \cdot \prod_{p=1}^{P-1} \exp(F(\mathbf{x}, y_{k-p} = \mathbf{Nil})) \cdot \sum_{u_{k-P}} \alpha(u_{k-P}, k-P)$$

Inference: Backward Algorithm

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

Eli	*New*	*win*	*World*
Eli Manning	*New York*		*World Series*
Manning	*York*		*Series*
	New York Giants		
	Giants		

$\beta(u_k, k)$

Inference: Backward Algorithm

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

Eli | *New* | *win* | *World*

Eli Manning | *New York* | *World Series*

Manning | *New York Giants* | *Series*

York

Giants

$$\beta(u_k, k) = \sum_{u_{k+Q}} \exp(F(\mathbf{x}, y_{k+Q} = u_{k+Q})) \cdot \prod_{q=1}^{Q-1} \exp(F(\mathbf{x}, y_{k+q} = \mathbf{Nil})) \cdot \beta(u_{k+Q}, k+Q)$$

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]

Data	#Tweet	#Entity	Date
NEEL Train	2,340	2,202	Jul. ~ Aug. 11
NEEL Test	1,164	687	Jul. ~ Aug. 11
TACL-IE	500	300	Dec. 12
TACL-IR	980	-	Dec. 12

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

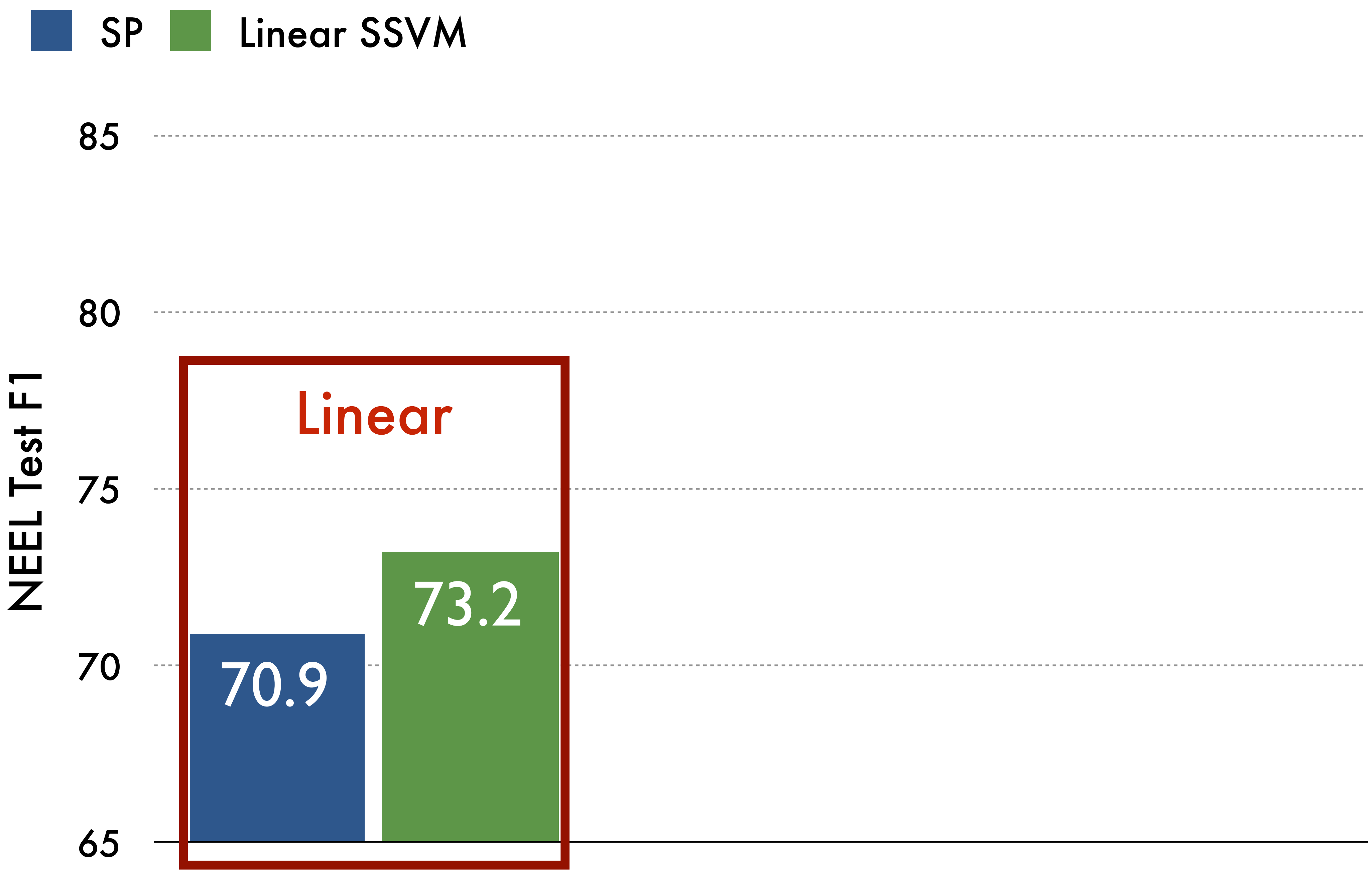
	Structured	Non-linear	Tree-based
Structured Perceptron	✓		
Linear SSVM*	✓		
Polynomial SSVM	✓	✓	
LambdaRank		✓	
MART#		✓	✓
S-MART	✓	✓	✓

* previous state of the art system

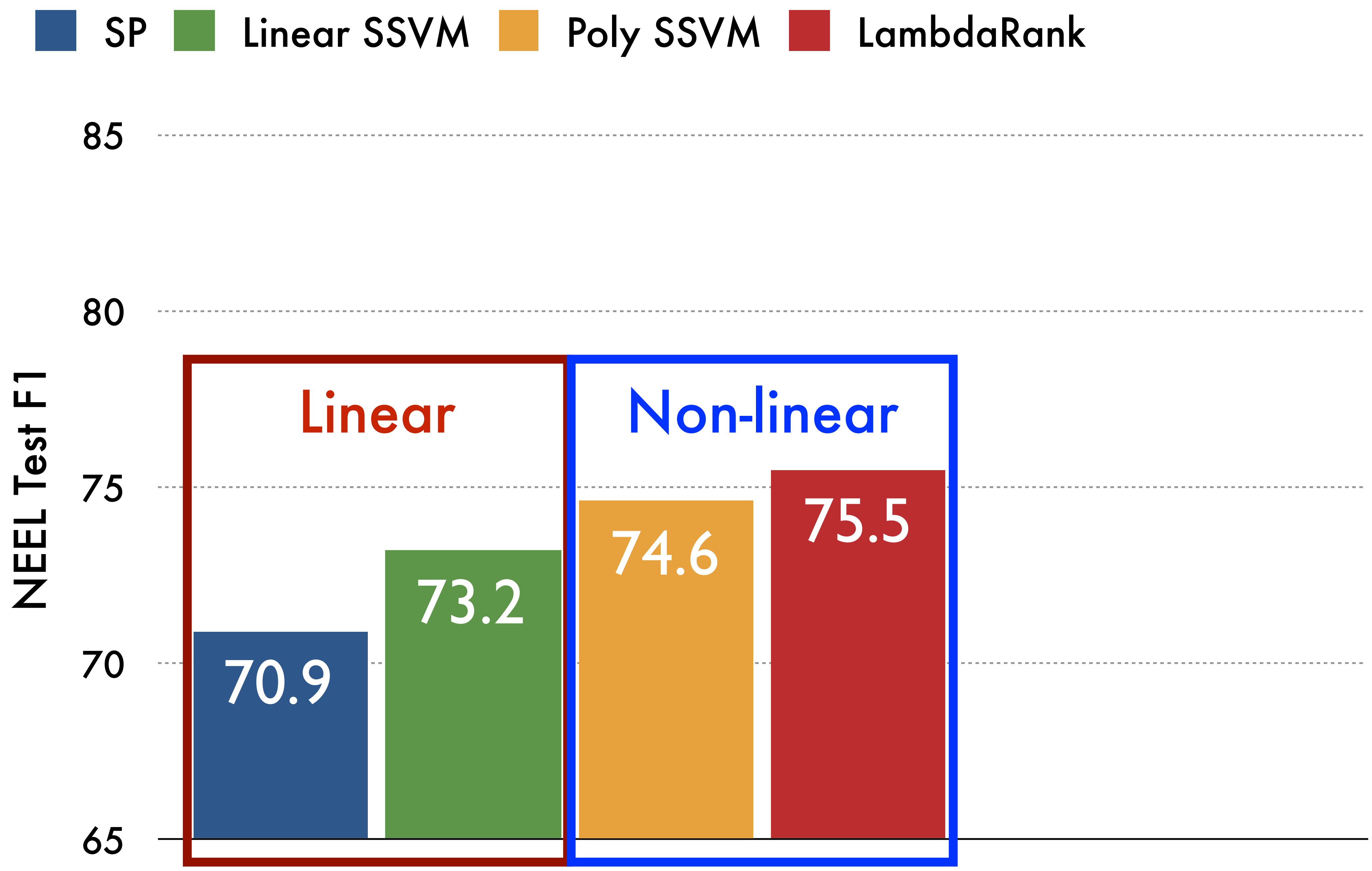
winning system of NEEL challenge 2014

IE-driven Evaluation

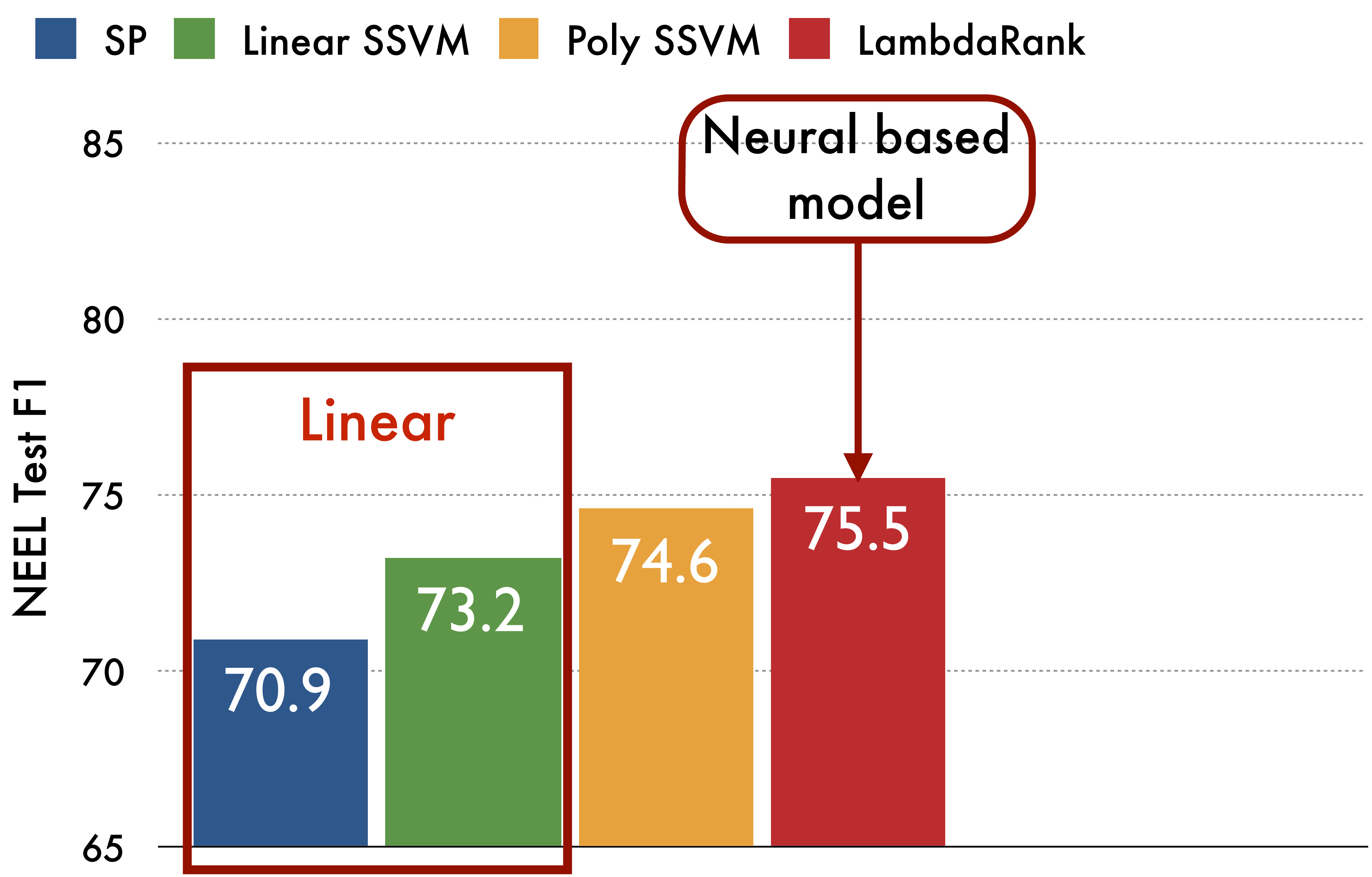
IE-driven Evaluation



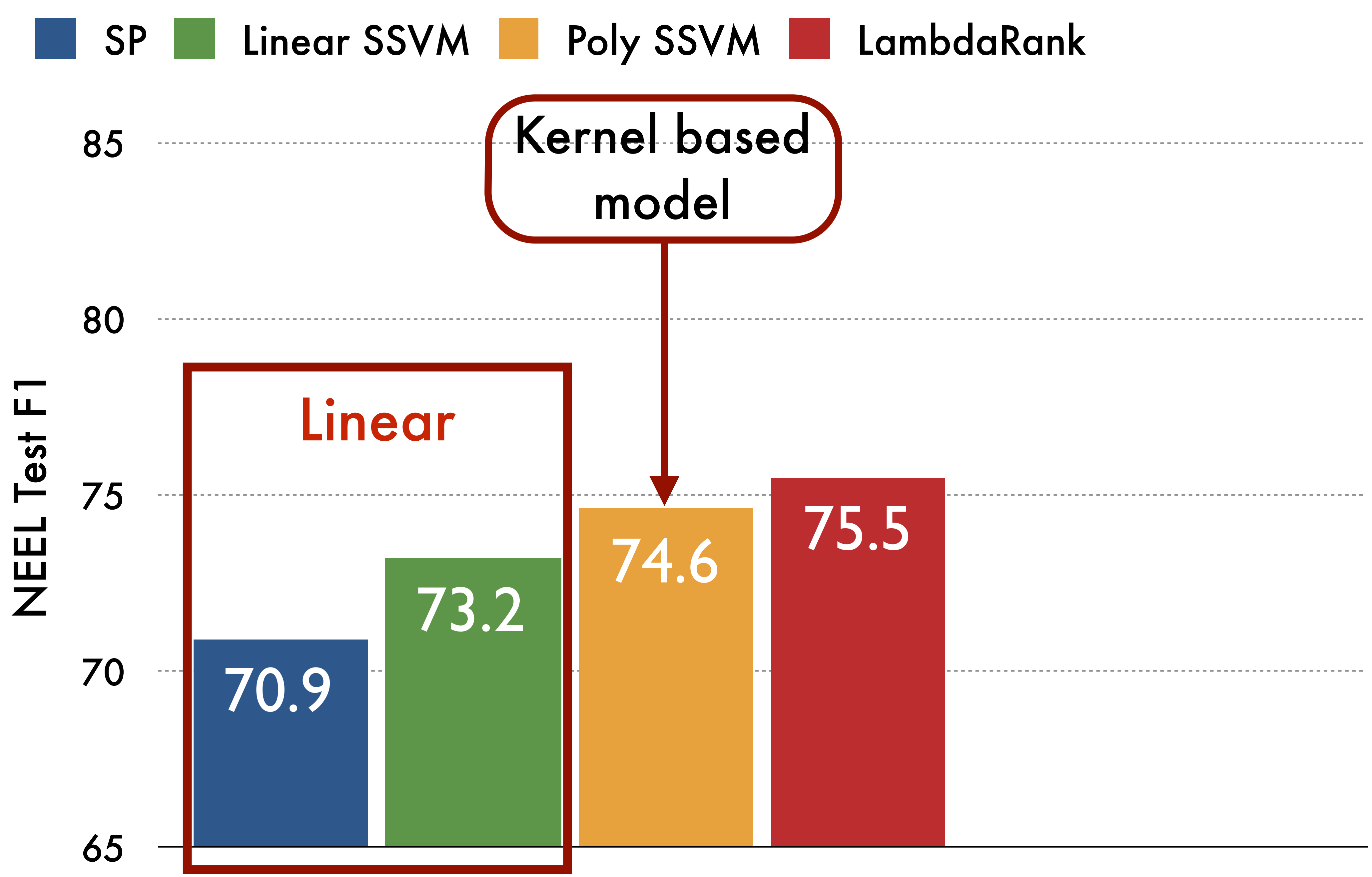
IE-driven Evaluation



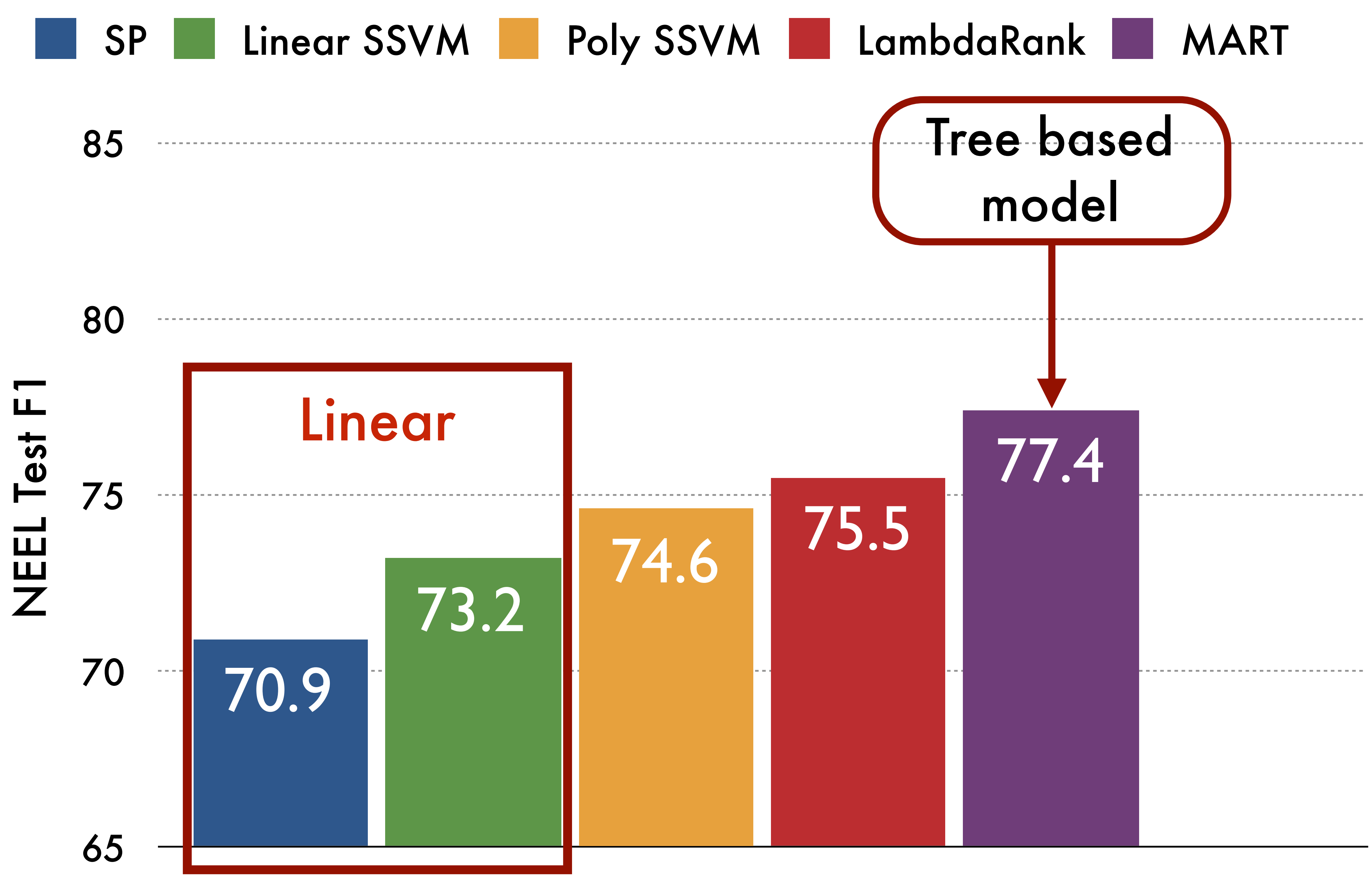
IE-driven Evaluation



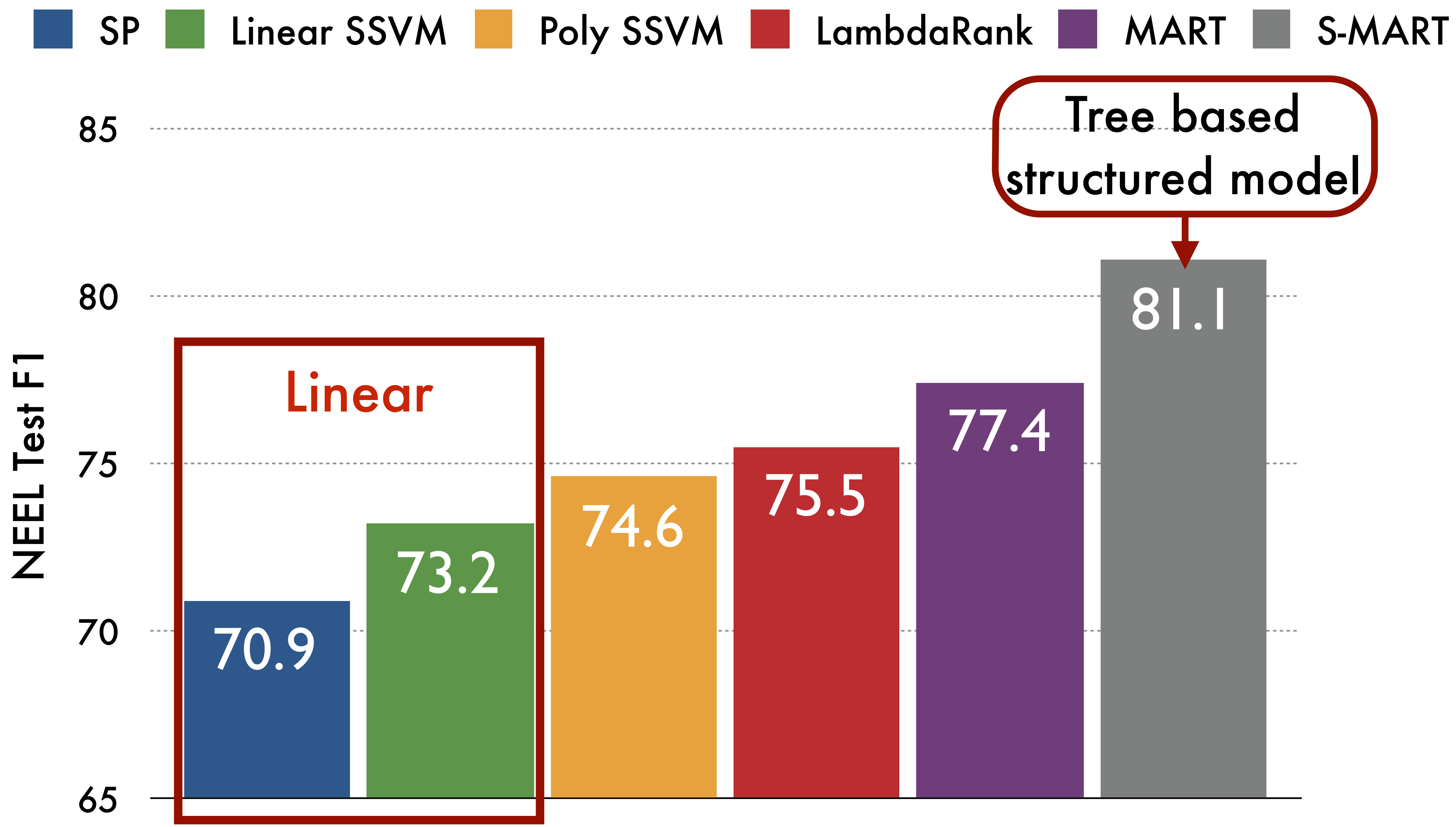
IE-driven Evaluation



IE-driven Evaluation

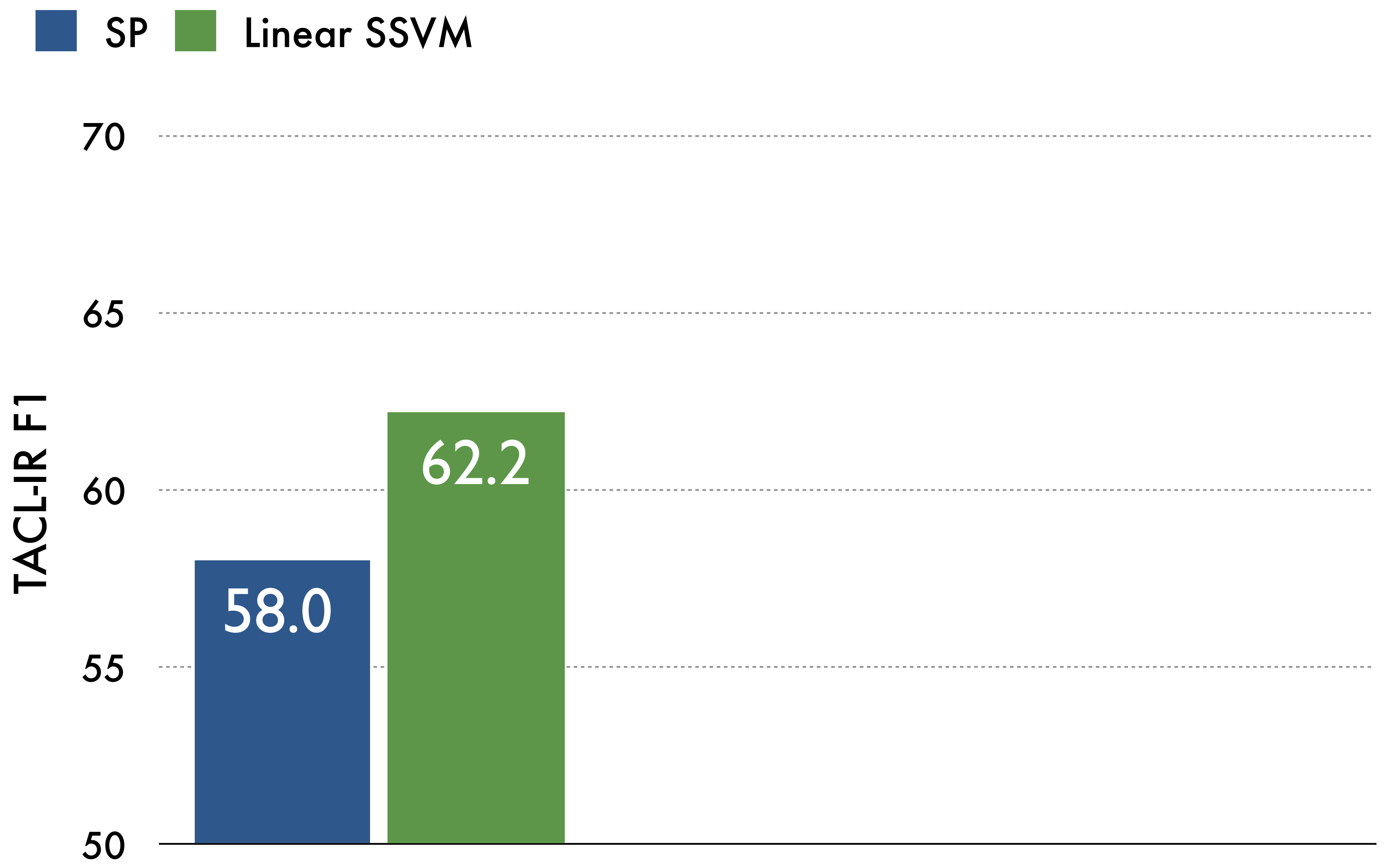


IE-driven Evaluation

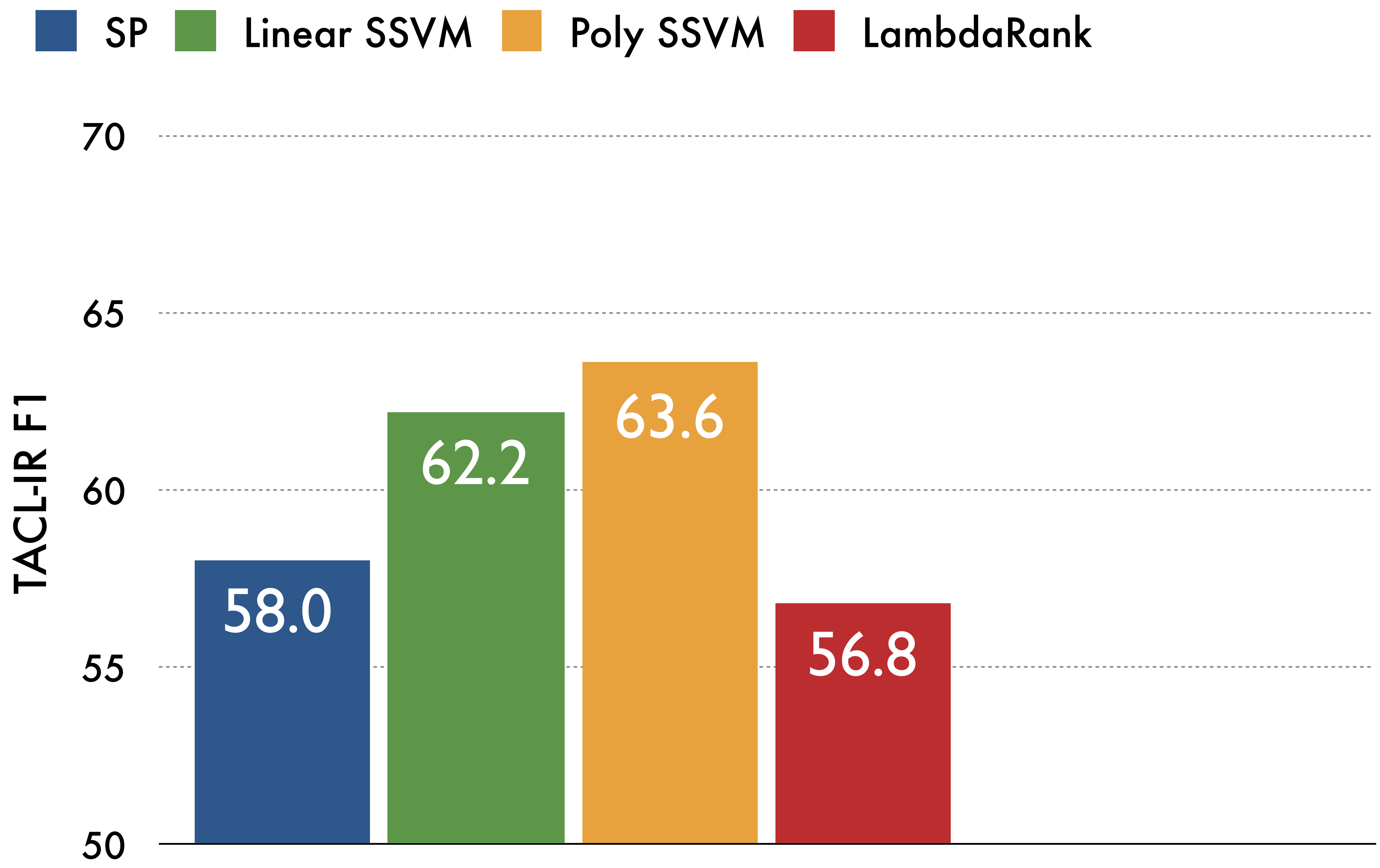


IR-driven Evaluation

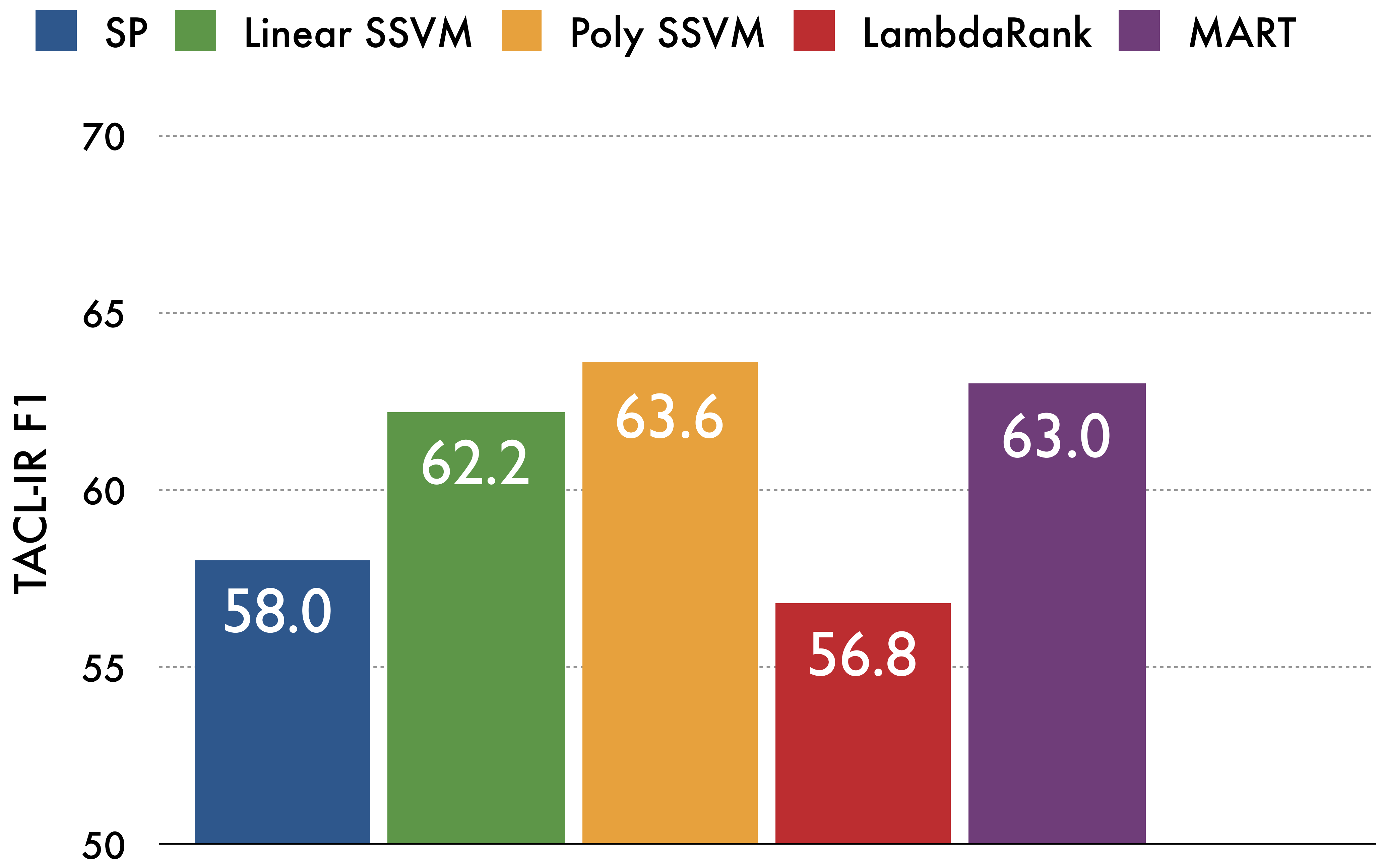
IR-driven Evaluation



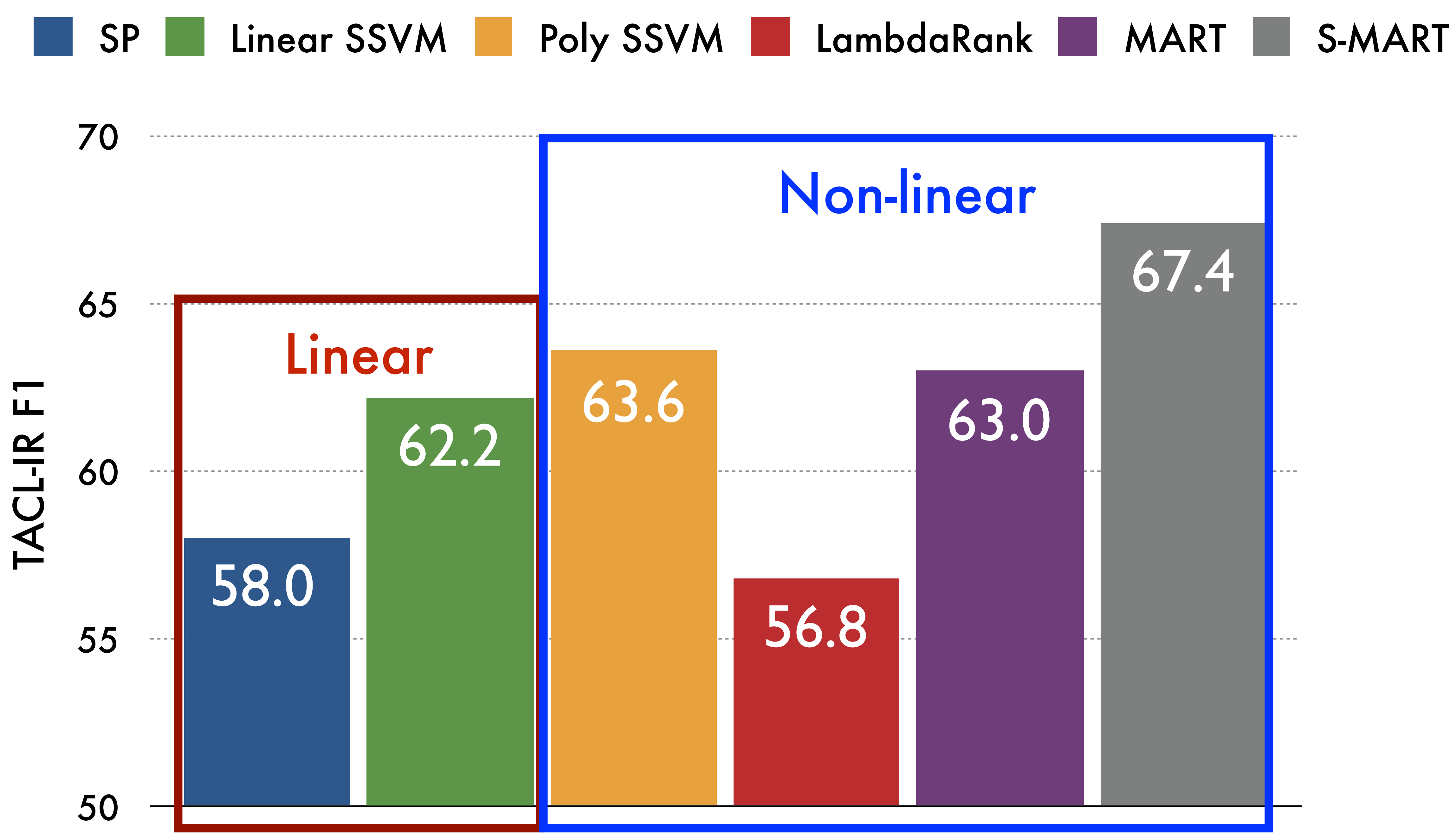
IR-driven Evaluation



IR-driven Evaluation



IR-driven Evaluation



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!