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

Yi Yang
School of Interactive Computing
Georgia Institute of Technology
yiyang@gatech.edu

Ming-Wei Chang
Microsoft Research
minchang@microsoft.com

Original version: Proceedings of ACL 2015, pp. 504-513. The experiments in the original version is done on the subset of the datasets. This revision of Sep 2016 includes updated statistics for the actual datasets used in the original version and additional results for the full datasets (in appendix). While the experimental results are largely similar, we update the paper for the sake of completeness.

Abstract

Non-linear models recently receive a lot of attention as people are starting to discover the power of statistical and embedding features. However, tree-based models are seldom studied in the context of structured learning despite their recent success on various classification and ranking tasks. In this paper, we propose S-MART, a tree-based structured learning framework based on multiple additive regression trees. S-MART is especially suitable for handling tasks with dense features, and can be used to learn many different structures under various loss functions.

We apply S-MART to the task of tweet entity linking — a core component of tweet information extraction, which aims to identify and link name mentions to entities in a knowledge base. A novel inference algorithm is proposed to handle the special structure of the task. The experimental results show that S-MART significantly outperforms state-of-the-art tweet entity linking systems.

1 Introduction

Many natural language processing (NLP) problems can be formalized as structured prediction tasks. Standard algorithms for structured learning include Conditional Random Field (CRF) (Lafferty et al., 2001) and Structured Supported Vector Machine (SSVM) (Tsochantaridis et al., 2004). These algorithms, usually equipped with a linear model and sparse lexical features, achieve state-of-the-art performances in many NLP applications such as part-of-speech tagging, named entity recognition and dependency parsing.

This classical combination of linear models and sparse features is challenged by the recent emerging usage of dense features such as statistical and embedding features. Tasks with these low dimensional dense features require models to be more sophisticated to capture the relationships between features. Therefore, non-linear models start to receive more attention as they are often more expressive than linear models.

Tree-based models such as boosted trees (Friedman, 2001) are flexible non-linear models. They can handle categorical features and count data better than other non-linear models like Neural Networks. Unfortunately, to the best of our knowledge, little work has utilized tree-based methods for structured prediction, with the exception of TreeCRF (Dietterich et al., 2004).

In this paper, we propose a novel structured learning framework called S-MART (**S**tructured **M**ultiple **A**dditive **R**egression **T**rees). Unlike TreeCRF, S-MART is very versatile, as it can be applied to tasks beyond sequence tagging and can be trained under various objective functions. S-MART is also powerful, as the high order relationships between features can be captured by non-linear regression trees.

We further demonstrate how S-MART can be applied to tweet entity linking, an important and challenging task underlying many applications including product feedback (Asur and Huberman, 2010) and topic detection and tracking (Mathioudakis and Koudas, 2010). We apply S-MART to

entity linking using a simple logistic function as the loss function and propose a novel inference algorithm to prevent overlaps between entities.

Our contributions are summarized as follows:

- We propose a novel structured learning framework called S-MART. S-MART combines non-linearity and efficiency of tree-based models with structured prediction, leading to a family of new algorithms. (Section 2)
- We apply S-MART to tweet entity linking. Building on top of S-MART, we propose a novel inference algorithm for non-overlapping structure with the goal of preventing conflicting entity assignments. (Section 3)
- We provide a systematic study of evaluation criteria in tweet entity linking by conducting extensive experiments over major data sets. The results show that S-MART significantly outperforms state-of-the-art entity linking systems, including the system that is used to win the NEEL 2014 challenge (Cano and others, 2014). (Section 4)

2 Structured Multiple Additive Regression Trees

The goal of a structured learning algorithm is to learn a joint scoring function S between an input $\mathbf{x}$ and an output structure $\mathbf{y}$, $S : (\mathbf{x}, \mathbf{y}) \rightarrow \mathbb{R}$. The structured output $\mathbf{y}$ often contains many interdependent variables, and the number of the possible structures can be exponentially large with respect to the size of $\mathbf{x}$. At test time, the prediction $\mathbf{y}$ for $\mathbf{x}$ is obtained by

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

where $Gen(\mathbf{x})$ represents the set of all valid output structures for $\mathbf{x}$.

Standard learning algorithms often directly optimize the model parameters. For example, assume that the joint scoring function S is parameterized by θ. Then, gradient descent algorithms can be used to optimize the model parameters θ iteratively. More specifically,

$$\theta_m = \theta_{m-1} - \eta_m \frac{\partial L(\mathbf{y}^*, S(\mathbf{x}, \mathbf{y}; \theta))}{\partial \theta_{m-1}}, \quad (1)$$

where $\mathbf{y}^*$ is the gold structure, $L(\mathbf{y}^*, S(\mathbf{x}, \mathbf{y}; \theta))$ is a loss function and η_m is the learning rate of the m-th iteration.

In this paper, we propose a framework called **Structured Multiple Additive Regression Trees (S-MART)**, which generalizes Multiple Additive Regression Trees (MART) for structured learning problems. Different from Equation (1), S-MART does *not* directly optimize the model parameters; instead, it approximates the optimal scoring function that minimize the loss by adding (weighted) regression tree models iteratively.

Due to the fact that there are exponentially many input-output pairs in the training data, S-MART assumes that the joint scoring function can be decomposed as

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

where $\Omega(\mathbf{x})$ contains the set of the all factors for input $\mathbf{x}$ and $\mathbf{y}_k$ is the sub-structure of $\mathbf{y}$ that corresponds to the k-th factor in $\Omega(\mathbf{x})$. For instance, in the task of word alignment, each factor can be defined as a pair of words from source and target languages respectively. Note that we can recover $\mathbf{y}$ from the union of $\{\mathbf{y}_k\}_1^K$.

The factor scoring function $F(\mathbf{x}, \mathbf{y}_k)$ can be optimized by performing gradient descent in the function space in the following manner:

$$F_m(\mathbf{x}, \mathbf{y}_k) = F_{m-1}(\mathbf{x}, \mathbf{y}_k) - \eta_m g_m(\mathbf{x}, \mathbf{y}_k) \quad (2)$$

where function $g_m(\mathbf{x}, \mathbf{y}_k)$ is the functional gradient.

Note that g_m is a *function* rather than a vector. Therefore, modeling g_m theoretically requires an infinite number of data points. We can address this difficulty by approximating g_m with a finite number of point-wise functional gradients

$$g_m(\mathbf{x}, \mathbf{y}_k = u_k) = \left[\frac{\partial L(\mathbf{y}^*, S(\mathbf{x}, \mathbf{y}_k = u_k))}{\partial F(\mathbf{x}, \mathbf{y}_k = u_k)} \right]_{F(\mathbf{x}, \mathbf{y}_k) = F_{m-1}(\mathbf{x}, \mathbf{y}_k)} \quad (3)$$

where u_k index a valid sub-structure for the k-th factor of $\mathbf{x}$.

The key point of S-MART is that it approximates $-g_m$ by modeling the point-wise negative functional gradients using a regression tree h_m. Then the factor scoring function can be obtained by

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

Algorithm 1 S-MART: A family of structured learning algorithms with multiple additive regression trees

1: $F_0(\mathbf{x}, \mathbf{y}_k) = 0$
2: **for** $m = 1$ to M **do**: ▷ going over all trees
3: $D \leftarrow \emptyset$
4: **for** all examples **do**: ▷ going over all examples
5: **for** $\mathbf{y}_k \in \Omega(\mathbf{x})$ **do**: ▷ going over all factors
6: For all u_k, obtain g_{ku} by Equation (3)
7: $D \leftarrow D \cup \{(\Phi(\mathbf{x}, \mathbf{y}_k = u_k), -g_{ku})\}$
8: **end for**
9: **end for**
10: $h_m(\mathbf{x}, \mathbf{y}_k) \leftarrow \texttt{TrainRegressionTree}(D)$
11: $F_m(\mathbf{x}, \mathbf{y}_k) = F_{m-1}(\mathbf{x}, \mathbf{y}_k) + h_m(\mathbf{x}, \mathbf{y}_k)$
12: **end for**

where $h_m(\mathbf{x}, \mathbf{y}_k)$ is also called a basis function and η_m can be simply set to 1 (Murphy, 2012).

The detailed S-MART algorithm is presented in Algorithm 1. The factor scoring function $F(\mathbf{x}, \mathbf{y}_k)$ is simply initialized to zero at first (line 1). After this, we iteratively update the function by adding regression trees. Note that the scoring function is shared by all the factors. Specifically, given the current decision function F_{m-1}, we can consider line 3 to line 9 a process of generating the pseudo training data D for modeling the regression tree. For each training example, S-MART first computes the point-wise functional gradients according to Equation (3) (line 6). Here we use g_{ku} as the abbreviation for $g_m(\mathbf{x}, \mathbf{y}_k = u_k)$. In line 7, for each sub-structure u_k, we create a new training example for the regression problem by the feature vector $\Phi(\mathbf{x}, \mathbf{y}_k = u_k)$ and the negative gradient $-g_{ku}$. In line 10, a regression tree is constructed by minimizing differences between the prediction values and the point-wise negative gradients. Then a new basis function (modeled by a regression tree) will be added into the overall F (line 11).

It is crucial to note that S-MART is *a family of algorithms* rather than a single algorithm. S-MART is flexible in the choice of the loss functions. For example, we can use either logistic loss or hinge loss, which means that S-MART can train probabilistic models as well as non-probabilistic ones. Depending on the choice of factors, S-MART can handle various structures such as linear chains, trees, and even the semi-Markov chain (Sarawagi and Cohen, 2004).

S-MART versus MART There are two key differences between S-MART and MART. First, S-MART decomposes the joint scoring function $S(\mathbf{x}, \mathbf{y})$ into factors to address the problem of the exploding number of input-output pairs for structured learning problems. Second, S-MART models a single scoring function $F(\mathbf{x}, \mathbf{y}_k)$ over inputs and output variables directly rather than O different functions $F^o(\mathbf{x})$, each of which corresponds to a label class.

S-MART versus TreeCRF TreeCRF can be viewed as a special case of S-MART, and there are two points where S-MART improves upon TreeCRF. First, the model designed in (Dietterich et al., 2004) is tailored for sequence tagging problems. Similar to MART, for a tagging task with O tags, they choose to model O functions $F^o(\mathbf{x}, o')$ instead of directly modeling the joint score of the factor. This imposes limitations on the feature functions, and TreeCRF is consequently unsuitable for many tasks such as entity linking.[1] Second, S-MART is more general in terms of the objective functions and applicable structures. In the next section, we will see how S-MART can be applied to a non-linear-chain structure and various loss functions.

3 S-MART for Tweet Entity Linking

We first formally define the task of tweet entity linking. As *input*, we are given a tweet, an entity database (*e.g.*, Wikipedia where each article is an entity), and a lexicon[2] which maps a surface form into a set of entity candidates. For each incoming tweet, all n-grams of this tweet will be used to find matches in the lexicon, and each match will form a mention candidate. As *output*, we map every mention candidate (*e.g.*, "new york giants") in the message to an entity (*e.g.*, NEW YORK GIANTS) or to **Nil** (*i.e.*, a non-entity). A mention candidate can often potentially link to multiple entities, which we call possible *entity assignments*.

This task is a structured learning problem, as the final entity assignments of a tweet should not overlap with each other.[3] We decompose this learning problem as follows: we make each mention candidate a factor, and the score of the entity assignments of a tweet is the sum of the score of each entity and mention candidate pair. Although all mention candidates are decomposed, the non-

[1] For example, entity linking systems need to model the similarity between an entity and the document. The TreeCRF formulation does not support such features.

[2] We use the standard techniques to construct the lexicon from anchor texts, redirect pages and other information resources.

[3] We follow the common practice and do not allow embedded entities.

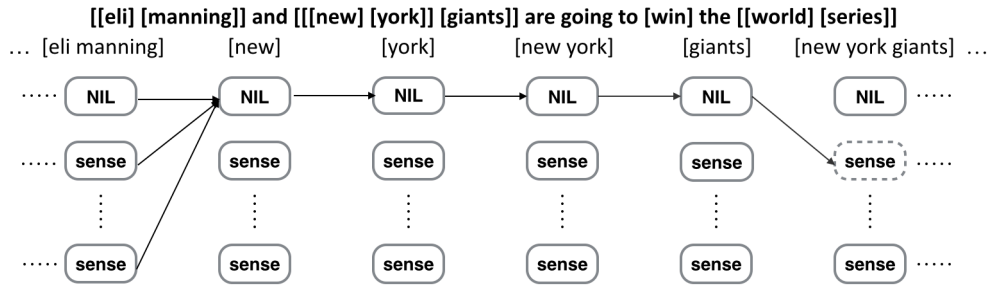

Figure 1: Example tweet and its mention candidates. Each mention candidate is marked as a pair of brackets in the original tweet and forms a column in the graph. The graph demonstrates the non-overlapping constraint. To link the mention candidate "new york giants" to a non-**Nil** entity, the system has to link previous four overlapping mention candidates to **Nil**. The mention candidate "eli manning" is not affected by "new york giants". **Note that this is not a standard linear chain problem.**

overlapping constraint requires the system to perform global inference.

Consider the example tweet in Figure 1, where we show the tweet with the mention candidates in brackets. To link the mention candidate "new york giants" to a non-**Nil** entity, the system has to link previous overlapping mention candidates to **Nil**. It is important to note that this is **not** a linear chain problem because of the non-overlapping constraint, and the inference algorithm needs to be carefully designed.

3.1 Applying S-MART

We derive specific model for tweet entity linking task with S-MART and use logistic loss as our running example. The hinge loss version of the model can be derived in a similar way.

Note that the tweet and the mention candidates are given. Let x be the tweet, u_k be the entity assignment of the k-th mention candidate. We use function $F(\mathbf{x}, y_k = u_k)$ to model the score of the k-th mention candidate choosing entity u_k.[4] The overall scoring function can be decomposed as follows:

$$S(\mathbf{x}, \mathbf{y} = \{u_k\}_{k=1}^K) = \sum_{k=1}^{K} F(\mathbf{x}, y_k = u_k)$$

S-MART utilizes regression trees to model the scoring function $F(\mathbf{x}, y_k = u_k)$, which requires point-wise functional gradient for each entity of every mention candidate. Let's first write down the logistic loss function as

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

where $Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp(S(\mathbf{x}, \mathbf{y}))$ is the potential function. Then the point-wise gradients can be computed as

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

where $\mathbf{1}[\cdot]$ represents an indicator function. The conditional probability $P(y_k = u_k|\mathbf{x})$ can be computed by a variant of the forward-backward algorithm, which we will detail in the next subsection.

3.2 Inference

The non-overlapping structure is distinct from linear chain and semi-Markov chain (Sarawagi and Cohen, 2004) structures. Hence, we propose a carefully designed forward-backward algorithm to calculate $P(y_k = u_k|\mathbf{x})$ based on current scoring function $F(\mathbf{x}, y_k = u_k)$ given by the regression trees. The non-overlapping constraint distinguishes our inference algorithm from other forward-backward variants.

To compute the forward probability, we sort[5] the mention candidates by their end indices and define forward recursion by

$$\begin{aligned} \alpha(u_1, 1) =& \exp(F(\mathbf{x}, y_1 = u_1)) \\ \alpha(u_k, k) =& \exp(F(\mathbf{x}, y_k = u_k)) \\ & \cdot \prod_{p=1}^{P-1} \exp(F(\mathbf{x}, y_{k-p} = \textbf{Nil})) \\ & \cdot \sum_{u_{k-P}} \alpha(u_{k-P}, k-P) \end{aligned} \tag{4}$$

where $k - P$ is the index of the previous non-overlapping mention candidate. Intuitively, for

[4] Note that each mention candidate has different own entity sets.

[5] Sorting helps the algorithms find non-overlapping candidates.

the k-th mention candidate, we need to identify its nearest non-overlapping fellow and recursively compute the probability. The overlapping mention candidates can only take the **Nil** entity.

Similarly, we can sort the mention candidates by their start indices and define backward recursion by

$$\begin{aligned}\beta(u_K, K) =& 1 \\ \beta(u_k, k) =& \sum_{u_{k+Q}} \exp(F(\mathbf{x}, y_{k+Q} = u_{k+Q})) \\ & \cdot \prod_{q=1}^{Q-1} \exp(F(\mathbf{x}, y_{k+q} = \textbf{Nil})) \\ & \cdot \beta(u_{k+Q}, k+Q) \end{aligned} \quad (5)$$

where $k + Q$ is the index of the next non-overlapping mention candidate. Note that the third terms of equation (4) or (5) will vanish if there are no corresponding non-overlapping mention candidates.

Given the potential function can be computed by $Z(\mathbf{x}) = \sum_{u_k} \alpha(u_k, k)\beta(u_k, k)$, for entities that are not **Nil**,

$$\begin{aligned}P(y_k = u_k|\mathbf{x}) =& \frac{\exp(F(\mathbf{x}, y_k = u_k)) \cdot \beta(u_k, k)}{Z(\mathbf{x})} \\ & \cdot \prod_{p=1}^{P-1} \exp(F(\mathbf{x}, y_{k-p} = \textbf{Nil})) \\ & \cdot \sum_{u_{k-P}} \alpha(u_{k-P}, k-P) \end{aligned} \quad (6)$$

The probability for the special token **Nil** can be obtained by

$$P(y_k = \textbf{Nil}|\mathbf{x}) = 1 - \sum_{u_k \neq \textbf{Nil}} P(y_k = u_k|\mathbf{x}) \quad (7)$$

In the worst case, the total cost of the forward-backward algorithm is $\mathcal{O}(\max\{TK, K^2\})$, where T is the number of entities of a mention candidate.[6]

Finally, at test time, the decoding problem $\arg\max_{\mathbf{y}} S(\mathbf{x}, \mathbf{y})$ can be solved by a variant of the Viterbi algorithm.

3.3 Beyond S-MART: Modeling entity-entity relationships

It is important for entity linking systems to take advantage of the entity-to-entity information while making local decisions. For instance, the identification of entity "eli manning" leads to a strong clue for linking "new york giants" to the NFL team.

Instead of defining a more complicated structure and learning everything jointly, we employ a two-stage approach as the solution for modeling entity-entity relationships after we found that S-MART achieves high precision and reasonable recall. Specifically, in the first stage, the system identifies all possible entities with basic features, which enables the extraction of entity-entity features. In the second stage, we re-train S-MART on a union of basic features and entity-entity features. We define entity-entity features based on the Jaccard distance introduced by Guo et al. (2013).

Let $\Gamma(e_i)$ denotes the set of Wikipedia pages that contain a hyperlink to an entity e_i and $\Gamma(t_{-i})$ denotes the set of pages that contain a hyperlink to any identified entity e_j of the tweet t in the first stage excluding e_i. The Jaccard distance between e_i and t is

$$Jac(e_i, t) = \frac{|\Gamma(e_i) \cap \Gamma(t_{-i})|}{|\Gamma(e_i) \cup \Gamma(t_{-i})|}.$$

In addition to the Jaccard distance, we add one additional binary feature to indicate if the current entity has the highest Jaccard distance among all entities for this mention candidate.

4 Experiments

Our experiments are designed to answer the following three research questions in the context of tweet entity linking:

- Do non-linear learning algorithms perform better than linear learning algorithms?
- Do structured entity linking models perform better than non-structured ones?
- How can we best capture the relationships between entities?

4.1 Evaluation Methodology and Data

We evaluate each entity linking system using two evaluation policies: Information Extraction (IE) driven evaluation and Information Retrieval (IR) driven evaluation. For both evaluation settings, precision, recall and F1 scores are reported. Our data is constructed from two publicly available sources: Named Entity Extraction & Linking (NEEL) Challenge (Cano et al., 2014) datasets,

[6] The cost is $\mathcal{O}(K^2)$ only if every mention candidate of the tweet overlaps other mention candidates. In practice, the algorithm is nearly linear w.r.t K.

and the datasets released by Fang and Chang (2014). Note that we gather two datasets from Fang and Chang (2014) and they are used in two different evaluation settings. We refer to these two datasets as TACL-IE and TACL-IR, respectively. We perform some data cleaning and unification on these sets. The statistics of the datasets are presented in Table 1.[7]

Data	#Tweet	#Entity	Date
NEEL Train	1171	2202	Jul. ~Aug. 11
NEEL Test	398	687	Jul. ~Aug. 11
TACL-IE	180	300	Dec. 12
TACL-IR	980	NA	Dec. 12

Table 1: Statistics of data sets.

IE-driven evaluation The IE-driven evaluation is the standard evaluation for an end-to-end entity linking system. We follow Carmel et al. (2014) and relax the definition of the correct mention boundaries, as they are often ambiguous. A mention boundary is considered to be correct if it overlaps (instead of being the same) with the gold mention boundary. Please see (Carmel et al., 2014) for more details on the procedure of calculating the precision, recall and F1 score.

The NEEL and TACL-IE datasets have different annotation guidelines and different choices of knowledge bases, so we perform the following procedure to clean the data and unify the annotations. We first filter out the annotations that link to entities excluded by our knowledge base. We use the same knowledge base as the ERD 2014 competition (Carmel et al., 2014), which includes the union of entities in Wikipedia and Freebase. Second, we follow NEEL annotation guideline and re-annotate TACL-IE dataset. For instance, in order to be consistent with NEEL, all the user tags (e.g. @BarackObama) are re-labeled as entities in TACL-IE.

We train all the models with NEEL Train dataset and evaluate different systems on NEEL Test and TACL-IE datasets. In addition, we sample 800 tweets from NEEL Train dataset as our development set to perform parameter tuning.

IR-driven evaluation The IR-driven evaluation is proposed by Fang and Chang (2014). It is motivated by a key application of entity linking — retrieval of relevant tweets for target entities, which is crucial for downstream applications such as product research and sentiment analysis. In particular, given a query entity we can search for tweets based on the match with some potential surface forms of the query entity. Then, an entity linking system is evaluated by its ability to correctly identify the presence or absence of the query entity in every tweet. Our IR-driven evaluation is based on the TACL-IR set, which includes 980 tweets sampled for ten query entities of five entity types (roughly 100 tweets per entity). About 37% of the sampled tweets did not mention the query entity due to the anchor ambiguity.

4.2 Experimental Settings

Features We employ a total number of 37 dense features as our basic feature set. Most of the features are adopted from (Guo et al., 2013)[8], including various statistical features such as the probability of the surface to be used as anchor text in Wikipedia. We also add additional Entity Type features correspond to the following entity types: Character, Event, Product and Brand. Finally, we include several NER features to indicate each mention candidate belongs to one the following NER types: Twitter user, Twitter hashtag, Person, Location, Organization, Product, Event and Date.

Algorithms Table 2 summarizes all the algorithms that are compared in our experiments. First, we consider two linear structured learning algorithms: Structured Perceptron (Collins, 2002) and Linear Structured SVM (SSVM) (Tsochantaridis et al., 2004).

For non-linear models, we consider polynomial SSVM, which employs polynomial kernel inside the structured SVM algorithm. We also include LambdaRank (Burges et al., 2007), a neural-based learning to rank algorithm, which is widely used in the information retrieval literature. We further compare with MART, which is designed for performing multiclass classification using log loss without considering the structured information. Finally, we have our proposed log-loss S-MART algorithm, as described in Section 3. [9]

Note that our baseline systems are quite strong.

[7]The datasets can be downloaded from `http://research.microsoft.com/en-us/downloads/24c267d7-4c19-41e8-8de1-2c116fcbdbd3/default.aspx`. We exclude Twitter messages that contain no ground truth entity mentions in the main evaluation. The statistics of the full datasets and the corresponding results are available in Appendix A.

[8]We consider features of Base, Capitalization Rate, Popularity, Context Capitalization and Entity Type categories.

[9]Our pilot experiments show that the log-loss S-MART consistently outperforms the hinge-loss S-MART.

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

Table 2: Included algorithms and their properties.

Linear SSVM has been used in one of the state-of-the-art tweet entity linking systems (Guo et al., 2013), and the system based on MART is the winning system of the 2014 NEEL Challenge (Cano and others, 2014)[10].

Table 2 summarizes several properties of the algorithms. For example, most algorithms are structured (e.g. they perform dynamic programming at test time) except for MART and LambdaRank, which treat mention candidates independently.

Parameter tuning All the hyper-parameters are tuned on the development set. Then, we re-train our models on full training data (including the dev set) with the best parameters. We choose the soft margin parameter C from $\{0.5, 1, 5, 10\}$ for two structured SVM methods. After a preliminary parameter search, we fixed the number of trees to 300 and the minimum number of documents in a leaf to 30 for all tree-based models. For LambdaRank, we use a two layer feed forward network. We select the number of hidden units from $\{10, 20, 30, 40\}$ and learning rate from $\{0.1, 0.01, 0.001\}$.

It is widely known that F1 score can be affected by the trade-off between precision and recall. In order to make the comparisons between all algorithms fairer in terms of F1 score, we include a post-processing step to balance precision and recall for all the systems. Note the tuning is only conducted for the purpose of robust evaluation. In particular, we adopt a simple tuning strategy that works well for all the algorithms, in which we add a bias term b to the scoring function value of **Nil**:

$$F(\mathbf{x}, y_k = \textbf{Nil}) \leftarrow F(\mathbf{x}, y_k = \textbf{Nil}) + b.$$

We choose the bias term b from values between -3.0 to 3.0 on the dev set and apply the same bias term at test time.

[10]Note that the numbers we reported here are different from the results in NEEL challenge due to the fact that we have cleaned the datasets and the evaluation metrics are slightly different in this paper.

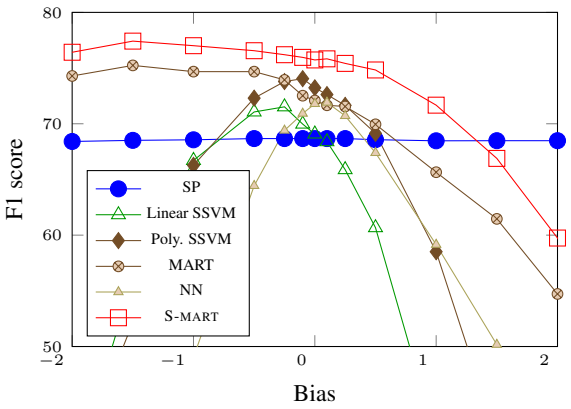

Figure 2: Balance precisions and recalls. X-axis corresponds to values of the bias terms for the special token **Nil**. Note that S-MART is still the overall winning system without tuning the threshold.

4.3 Results

Table 3 presents the empirical findings for S-MART and competitive methods on tweet entity linking task in both IE and IR settings. In the following, we analyze the empirical results in details.

Linear models vs. non-linear models Table 3 clearly shows that linear models perform worse than non-linear models when they are restricted to the IE setting of the tweet entity linking task. The story is similar in IR-driven evaluation, with the exception of LambdaRank. Among the linear models, linear SSVM demonstrates its superiority over Structured Perceptron on all datasets, which aligns with the results of (Tsochantaridis et al., 2005) on the named entity recognition task.

We have many interesting observations on the non-linear models side. First, by adopting a polynomial kernel, the non-linear SSVM further improves the entity linking performances on the NEEL datasets and TACL-IR dataset. Second, LambdaRank, a neural network based model, achieves better results than linear models in IE-driven evaluation, but the results in IR-driven evaluation are worse than all the other methods. We believe the reason for this dismal performance is that the neural-based method tends to overfit the IR setting given the small number of training examples. Third, both MART and S-MART significantly outperform alternative linear and non-linear methods in IE-driven evaluation and performs better or similar to other methods in IR-driven evaluation. This suggests that tree-based non-linear models are suitable for tweet entity linking task. Finally, S-MART outperforms previous state-of-

Model	NEEL Dev			NEEL Test			TACL-IE			TACL-IR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Structured Perceptron	75.8	62.8	68.7	79.1	64.3	70.9	74.4	63.0	68.2	86.2	43.8	58.0
Linear SSVM	78.0	66.1	71.5	80.5	67.1	73.2	**78.2**	64.7	70.8	86.7	48.5	62.2
Polynomial SSVM	77.7	70.7	74.0	81.3	69.0	74.6	76.8	64.0	69.8	91.1	48.8	63.6
LambdaRank	75.0	69.0	71.9	80.3	71.2	75.5	77.8	66.7	71.8	85.8	42.4	56.8
MART	76.2	74.3	75.2	76.8	78.0	77.4	73.4	71.0	72.2	**98.1**	46.4	63.0
S-MART	**79.1**	**75.8**	**77.4**	**83.2**	**79.2**	**81.1**	76.8	**73.0**	**74.9**	95.1	**52.2**	**67.4**
+ entity-entity	79.2	75.8	77.5	81.5	76.4	78.9	77.3	73.7	75.4	95.5	56.7	71.1

Table 3: IE-driven and IR-driven evaluation results for different models. The best results with basic features are in **bold**. The results are underlined if adding entity-entity features gives the overall best results. Please see Appendix.

the-art method Structured SVM by a surprisingly large margin. In the NEEL Test dataset, the difference is more than 10% F1. Overall, the results show that the shallow linear models are not expressive enough to capture the complex patterns in the data, which are represented by a few dense features.

Structured learning models To showcase structured learning technique is crucial for entity linking with non-linear models, we compare S-MART against MART directly. As shown in Table 3, S-MART can achieve higher precision and recall points compared to MART on all datasets in terms of IE-driven evaluation, and can improve F1 by 4 points on NEEL Test and TACL-IR datasets. The task of entity linking is to produce non-overlapping entity assignments that match the gold mentions. By adopting structured learning technique, S-MART is able to automatically take into account the non-overlapping constraint during learning and inference, and produce global optimal entity assignments for mention candidates of a tweet. One effect is that S-MART can easily eliminate some common errors caused by popular entities (e.g. new york in Figure 1).

Modeling entity-entity relationships Entity-entity relationships provide strong clues for entity disambiguation. In this paper, we use the simple two-stage approach described in Section 3.3 to capture the relationships between entities. As shown in Table 3, the significant improvement in IR-driven evaluation indicates the importance of incorporating entity-entity information.

Interestingly, while IR-driven results are significantly improved, IE-driven results are similar or even worse given entity-entity features. We believe the reason is that IE-driven and IR-driven evaluations focus on different aspects of tweet entity linking task. As Guo et al. (2013) shows that most mentions in tweets should be linked to the most popular entities, IE setting actually pays more attention on mention detection sub-problem. In contrast to IE setting, IR setting focuses on entity disambiguation, since we only need to decide whether the tweet is relevant to the query entity. Therefore, we believe that both evaluation policies are needed for tweet entity linking.

Balance Precision and Recall Figure 2 shows the results of tuning the bias term for balancing precision and recall on the dev set. The results show that S-MART outperforms competitive approaches without any tuning, with similar margins to the results after tuning. Balancing precision and recall improves F1 scores for all the systems, which suggests that the simple tuning method performs quite well. Finally, we have an interesting observation that different methods have various scales of model scores.

5 Related Work

Linear structured learning methods have been proposed and widely used in the literature. Popular models include Structured Perceptron (Collins, 2002), Conditional Random Field (Lafferty et al., 2001) and Structured SVM (Taskar et al., 2004; Tsochantaridis et al., 2005). Recently, many structured learning models based on neural networks have been proposed and are widely used in language modeling (Bengio et al., 2006; Mikolov et al., 2010), sentiment classification (Socher et al., 2013), as well as parsing (Socher et al., 2011). Cortes et al. (2014) recently proposed a boosting framework which treats different structured learning algorithms as base learners to ensemble structured prediction results.

Tree-based models have been shown to pro-

vide more robust and accurate performances than neural networks in some tasks of computer vision (Roe et al., 2005; Babenko et al., 2011) and information retrieval (Li et al., 2007; Wu et al., 2010), suggesting that it is worth to investigate tree-based non-linear models for structured learning problems. To the best of our knowledge, TreeCRF (Dietterich et al., 2004) is the only work that explores tree-based methods for structured learning problems. The relationships between TreeCRF and our work have been discussed in Section 2.[11]

Early research on entity linking has focused on well written documents (Bunescu and Pasca, 2006; Cucerzan, 2007; Milne and Witten, 2008). Due to the raise of social media, many techniques have been proposed or tailored to short texts including tweets, for the problem of entity linking (Ferragina and Scaiella, 2010; Meij et al., 2012; Guo et al., 2013) as well as the related problem of named entity recognition (NER) (Ritter et al., 2011). Recently, non-textual information such as spatial and temporal signals have also been used to improve entity linking systems (Fang and Chang, 2014). The task of entity linking has attracted a lot of attention, and many shared tasks have been hosted to promote entity linking research (Ji et al., 2010; Ji and Grishman, 2011; Cano and others, 2014; Carmel et al., 2014).

Building an end-to-end entity linking system involves in solving two interrelated sub-problems: mention detection and entity disambiguation. Earlier research on entity linking has been largely focused on the entity disambiguation problem, including most work on entity linking for well-written documents such as news and encyclopedia articles (Cucerzan, 2007) and also few for tweets (Liu et al., 2013). Recently, people have focused on building systems that consider mention detection and entity disambiguation jointly. For example, Cucerzan (2012) delays the mention detection decision and consider the mention detection and entity linking problem jointly. Similarly, Sil and Yates (2013) proposed to use a reranking approach to obtain overall better results on mention detection and entity disambiguation.

[11] Chen et al. (2015) independently propose a class of tree-based structured learning models using a similar formalization of S-MART. However, they focus on exploring second-order information of line chain structures, while we aim at handling different structures and objective functions.

6 Conclusion and Future Work

In this paper, we propose S-MART, a family of structured learning algorithms which is flexible on the choices of the loss functions and structures. We demonstrate the power of S-MART by applying it to tweet entity linking, and it significantly outperforms the current state-of-the-art entity linking systems. In the future, we would like to investigate the advantages and disadvantages between tree-based models and other non-linear models such as deep neural networks or recurrent neural networks.

Acknowledgments We thank the reviewers for their insightful feedback. We also thank Yin Li and Ana Smith for their valuable comments on earlier version of this paper.

References

S. Asur and B.A. Huberman. 2010. Predicting the future with social media. *arXiv preprint arXiv:1003.5699*.

Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. 2011. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 1619–1632.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186.

R. C Bunescu and M. Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the European Chapter of the ACL (EACL)*, pages 9–16.

Christopher JC Burges, Robert Ragno, and Quoc Le, Qu. 2007. Learning to rank with nonsmooth cost functions. In *Advances in neural information processing systems (NIPS)*, pages 193–200.

AE Cano et al. 2014. Microposts2014 neel challenge. In *Microposts2014 NEEL Challenge*.

Amparo E Cano, Giuseppe Rizzo, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie. 2014. Making sense of microposts (# microposts2014) named entity extraction & linking challenge. *Making Sense of Microposts (# Microposts2014)*.

David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang. 2014. Erd'14: entity recognition and disambiguation challenge. In *ACM SIGIR Forum*, pages 63–77.

Tianqi Chen, Sameer Singh, Ben Taskar, and Carlos Guestrin. 2015. Efficient second-order gradient boosting for conditional random fields. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 147–155.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the conference on Empirical methods in natural language processing (EMNLP)*, pages 1–8.

Corinna Cortes, Vitaly Kuznetsov, and Mehryar Mohri. 2014. Learning ensembles of structured prediction rules. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–716.

Silviu Cucerzan. 2012. The msr system for entity linking at tac 2012. In *Text Analysis Conference*.

Thomas G Dietterich, Adam Ashenfelter, and Yaroslav Bulatov. 2004. Training conditional random fields via gradient tree boosting. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 28–35.

Yuan Fang and Ming-Wei Chang. 2014. Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics (ACL)*, pages 259–272.

P. Ferragina and U. Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 1625–1628.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1020–1030.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1148–1158.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC)*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 282–289.

Ping Li, Qiang Wu, and Christopher J Burges. 2007. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems (NIPS)*, pages 897–904.

Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. 2013. Entity linking for tweets. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 1304–1311.

Michael Mathioudakis and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the ACM SIGMOD International Conference on Management of data (SIGMOD)*, pages 1155–1158.

E. Meij, W. Weerkamp, and M. de Rijke. 2012. Adding semantics to microblog posts. In *Proceedings of International Conference on Web Search and Web Data Mining (WSDM)*, pages 563–572.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048.

D. Milne and I. H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 509–518.

Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.

A. Ritter, S. Clark, Mausam, and O. Etzioni. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 1524–1534.

Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. 2005. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, pages 577–584.

Sunita Sarawagi and William W Cohen. 2004. Semimarkov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1185–1192.

Avirup Sil and Alexander Yates. 2013. Re-ranking for joint named-entity recognition and linking. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 2369–2374.

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 129–136.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

Ben Taskar, Carlos Guestrin, and Daphne Roller. 2004. Max-margin markov networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 104.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, pages 1453–1484.

Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval*, pages 254–270.

A Appendix: Results of Full Datasets

Data	#Tweet	#Entity	Date
NEEL Train	2340	2202	Jul. ˜Aug. 11
NEEL Test	1164	687	Jul. ˜Aug. 11
TACL-IE	500	300	Dec. 12

Table 4: Statistics of data sets introduced in Section 4.2.

Model	NEEL Test			TACL-IE		
	P	R	F1	P	R	F1
Structured Perceptron	74.1	59.5	66.0	59.4	54.7	56.9
Linear SSVM	72.3	65.5	68.8	61.0	63.0	62.0
Polynomial SSVM	68.0	**76.4**	72.0	58.4	66.0	62.0
MART	76.5	74.2	75.3	**61.5**	67.0	**64.1**
S-MART	**80.2**	75.4	**77.7**	60.1	**67.7**	63.6
+ entity-entity	82.1	72.8	77.2	67.1	67.3	67.2

Table 5: IE-driven evaluation results on the full NEEL Test and TACL-IE datasets for different models. The best results with basic features are in **bold**. The results are underlined if adding entity-entity features gives the overall best results.

The statistics of the full NEEL and TACL-IE datasets are shown in Table 4. Table 5 presents the IE-driven evaluation results on the datasets. On the NEEL Test dataset, non-linear models significantly outperform linear models, tree-based models perform much better than alternative methods, and our tree-based structured prediction method S-MART achieves the best results. The tree-based non-structured model MART yields best performance on the TACL-IE dataset with basic features, which slightly wins over S-MART by 0.5% F1. The entity-entity relationships improve the performance of S-MART on the TACL-IE dataset by 3.6 points of F1.