Collective Entity Disambiguation with Structured Gradient Tree Boosting

Yi Yang **Ozan Irsoy** **Kazi Shefaet Rahman**
Bloomberg LP
New York, NY 10022
{yyang464+oirsoy+krahman7}@bloomberg.net

Abstract

We present a gradient-tree-boosting-based structured learning model for jointly disambiguating named entities in a document. Gradient tree boosting is a widely used machine learning algorithm that underlies many top-performing natural language processing systems. Surprisingly, most works limit the use of gradient tree boosting as a tool for regular classification or regression problems, despite the structured nature of language. To the best of our knowledge, our work is the first one that employs the structured gradient tree boosting (SGTB) algorithm for collective entity disambiguation. By defining global features over previous disambiguation decisions and jointly modeling them with local features, our system is able to produce globally optimized entity assignments for mentions in a document. Exact inference is prohibitively expensive for our globally normalized model. To solve this problem, we propose Bidirectional Beam Search with Gold path (BiBSG), an approximate inference algorithm that is a variant of the standard beam search algorithm. BiBSG makes use of global information from both past and future to perform better local search. Experiments on standard benchmark datasets show that SGTB significantly improves upon published results. Specifically, SGTB outperforms the previous state-of-the-art neural system by near 1% absolute accuracy on the popular AIDA-CoNLL dataset.[1]

1 Introduction

Entity disambiguation (ED) refers to the process of linking an entity mention in a document to its corresponding entity record in a reference knowledge base (e.g., Wikipedia or Freebase). As a core information extraction task, ED plays an important role in the language understanding pipeline, underlying a variety of downstream applications such as relation extraction (Mintz et al., 2009; Riedel et al., 2010), knowledge base population (Ji and Grishman, 2011; Dredze et al., 2010), and question answering (Berant et al., 2013; Yih et al., 2015). This task is challenging because of the inherent ambiguity between mentions and the referred entities. Consider, for example, the mention 'Washington', which can be linked to a city, a state, a person, an university, or a lake (Fig. 1).

Fortunately, simple and effective features have been proposed to capture the ambiguity that are designed to model the similarity between a mention (and its local context) and a candidate entity, as well as the relatedness between entities that co-occur in a single document. These are typically statistical features estimated from entity-linked corpora, and similarity features that are pre-computed using distance metrics such as cosine. For example, a key feature for ED is the *prior probability* of an entity given a specific mention, which is estimated from mention-entity co-occurrence statistics. This simple feature alone can yield 70% to 80% accuracy on both news and Twitter texts (Lazic et al., 2015; Guo et al., 2013).

To capture the non-linear relationships between the low-dimensional dense features like statistical features, sophisticated machine learning models such as neural networks and gradient tree boosting are preferred over linear models. In particular, gradient tree boosting has been shown to be highly competitive for ED in recent work (Yang and Chang, 2015; Yamada et al., 2016). However, although achieving appealing results, existing gradient-tree-boosting-based ED systems typically operate on each individual mention, without attempting to jointly resolve entity mentions in a document together. Joint entity disambiguation has been shown to significantly boost performance when used in conjunction with other machine learning techniques (Ratinov et al., 2011; Hoffart et al., 2011). However, how to train a

[1] When ready, the code will be published at https://github.com/bloomberg/sgtb.

Proceedings of NAACL-HLT 2018, pages 777–786
New Orleans, Louisiana, June 1 - 6, 2018.

global gradient tree boosting model that produces coherent entity assignments for all the mentions in a document is still an open question.

In this work, we present, to the best of our knowledge, the first structured gradient tree boosting (SGTB) model for collective entity disambiguation. Building on the general SGTB framework introduced by Yang and Chang (2015), we develop a globally normalized model for ED that employs a conditional random field (CRF) objective (Lafferty et al., 2001). The model permits the utilization of global features defined between the current entity candidate and the entire decision history for previous entity assignments, which enables the global optimization for all the entity mentions in a document. As discussed in prior work (Smith and Johnson, 2007; Andor et al., 2016), globally normalized models are more expressive than locally normalized models.

As in many other global models, our SGTB model suffers from the difficulty of computing the partition function (normalization term) for training and inference. We adopt beam search to address this problem, in which we keep track of multiple hypotheses and sum over the paths in the beam. In particular, we propose Bidirectional Beam Search with Gold path (BiBSG) technique that is specifically designed for SGTB model training. Compared to standard beam search strategies, BiBSG reduces model variance and also enjoys the advantage in its ability to consider both past and future information when predicting an output.

Our contributions are:

- We propose a SGTB model for collectively disambiguating entities in a document. By jointly modeling local decisions and global structure, SGTB is able to produce globally optimal entity assignments for all the mentions.
- We present BiBSG, an efficient algorithm for approximate bidirectional inference. The algorithm is tailored to SGTB models, which can reduce model variance by generating more point-wise functional gradients for estimating the auxiliary regression models.
- SGTB achieves state-of-the-art (SOTA) results on various popular ED datasets, and it outperforms the previous SOTA systems by 1-2% absolute accuracy on the AIDA-CoNLL (Hoffart et al., 2011) dataset.

2 Model

In this section, we present a SGTB model for collective entity disambiguation. We first formally define the task of ED, and then describe a structured learning formalization for producing globally coherent entity assignments for mentions in a document. Finally, we show how to optimize the model using functional gradient descent.

For an input document, assume that we are given all the mentions of named entities within it. Also assume that we are given a lexicon that maps each mention to a set of entity candidates in a given reference entity database (e.g., Wikipedia or Freebase). The ED system maps each mention in the document to an entry in the entity database. Since a mention is often ambiguous on its own (i.e., the lexicon maps the mention to multiple entity candidates), the ED system needs to leverage two types of contextual information for disambiguation: local information based on the entity mention and its surrounding words, and global information that exploits the document-level coherence of the predicted entities. Note that modeling entity-entity coherence is very challenging, as the long-range dependencies between entities correspond to exponentially large search space.

We formalize this task as a structured learning problem. Let $\mathbf{x}$ be a document with T target mentions, and $\mathbf{y} = \{y_t\}_{t=1}^{T}$ be the entity assignments of the mentions in the document. We use $S(\mathbf{x}, \mathbf{y})$ to denote the joint scoring function between the input document and the output structure. In traditional NLP tasks, such as part-of-speech tagging and named entity recognition, we often rely on low-order Markov assumptions to decompose the global scoring function into a summation of local functions. ED systems, however, are often required to model nonlocal phenomena, as any pair of entities is potentially interdependent. Therefore, we choose the following decomposition:

$$S(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{T} F(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}), \qquad (1)$$

where $F(\mathbf{x}, y_t, \mathbf{y}_{1:t-1})$ is a factor scoring function. Specifically, a local prediction y_t depends on all the *previous decisions*, $\mathbf{y}_{1:t-1}$ in our model, which resembles recurrent neural network (RNN) models (Elman, 1990; Hochreiter and Schmidhuber, 1997).

We adopt a CRF loss objective, and define a

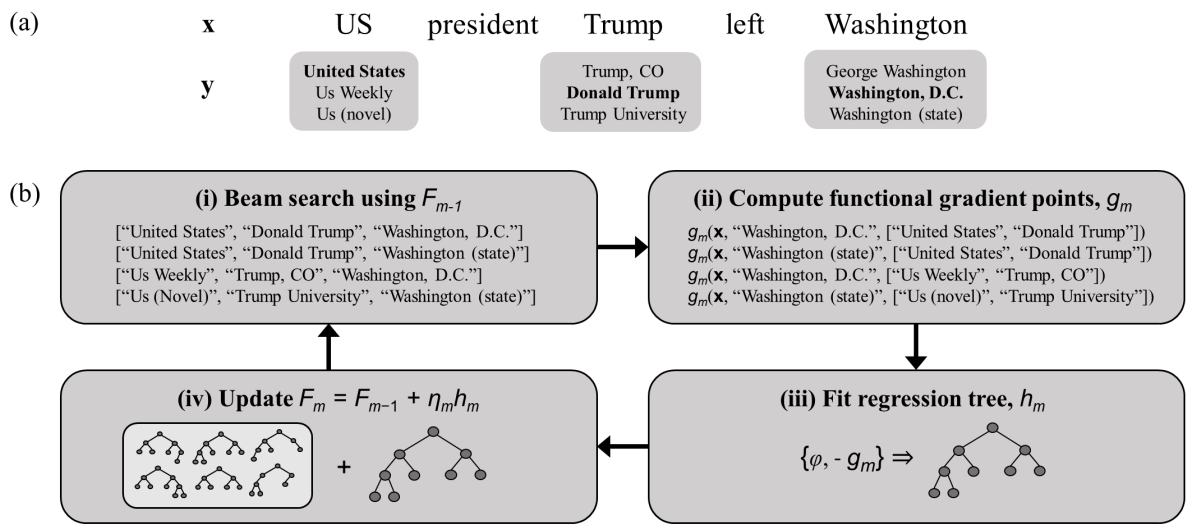

Figure 1: (a) Example document $\mathbf{x}$ with entity candidates for each mention (gold entities are in **bold**); (b) the m-th SGTB update iteration: (i) conduct beam search to sample candidate entity sequences (§ 3), (ii) compute point-wise functional gradients for each candidate sequence, (iii) fit a regression tree to the negative functional gradient points with input features, ϕ, (iv) update the factor scoring function, F, by adding the trained regression tree.

distribution over possible output structures as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp\{\sum_{t=1}^{T} F(\mathbf{x}, y_t, \mathbf{y}_{1:t-1})\}}{Z(\mathbf{x})}, \quad (2)$$

where

$$Z(\mathbf{x}) = \sum_{\mathbf{y}' \in Gen(\mathbf{x})} \exp\{\sum_{t=1}^{T} F(\mathbf{x}, y'_t, \mathbf{y}'_{1:t-1})\}$$

and $Gen(\mathbf{x})$ is the set of all possible sequences of entity assignments depending on the lexicon. $Z(\mathbf{x})$ is then a global normalization term. As shown in previous work, globally normalized models are very expressive, and also avoid the label bias problem (Lafferty et al., 2001; Andor et al., 2016). The inference problem is to find

$$\arg\max_{\mathbf{y} \in Gen(\mathbf{x})} p(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{y} \in Gen(\mathbf{x})} \sum_{t=1}^{T} F(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}). \quad (3)$$

2.1 Structured gradient tree boosting

An overview of our SGTB model is shown in Fig. 1. The model minimizes the negative log-likelihood of the data,

$$\begin{aligned} L(\mathbf{y}^*, S(\mathbf{x}, \mathbf{y})) &= -\log p(\mathbf{y}^*|\mathbf{x}) \\ &= \log Z(\mathbf{x}) - S(\mathbf{x}, \mathbf{y}^*), \end{aligned} \quad (4)$$

where $\mathbf{y}^*$ is the gold output structure.

In a standard CRF, the factor scoring function is typically assumed to have this form: $F(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}) = \theta^\top \phi(\mathbf{x}, y_t, \mathbf{y}_{1:t-1})$, where $\phi(\mathbf{x}, y_t, \mathbf{y}_{1:t-1})$ is the feature function and θ are the model parameters. The key idea of SGTB is that, instead of defining a parametric model and optimizing its parameters, we can directly optimize the factor scoring function $F(\cdot)$ iteratively by performing gradient descent in function space. In particular, suppose $F(\cdot) = F_{m-1}(\cdot)$ in the m-th iteration, we will update $F(\cdot)$ as follows:

$$\begin{aligned} F_m(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}) = F_{m-1}(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}) \\ - \eta_m g_m(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}), \end{aligned} \quad (5)$$

where

$$\begin{aligned} g_m(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}) &= \frac{\partial L(\mathbf{y}^*, S(\mathbf{x}, \mathbf{y}))}{\partial F(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}))} \\ &= p(\mathbf{y}_{1:t}|\mathbf{x}) - \mathbf{1}[\mathbf{y}_{1:t} = \mathbf{y}^*_{1:t}] \end{aligned} \quad (6)$$

is the functional gradient, η_m is the learning rate, and $\mathbf{1}[\cdot]$ represents an indicator function, which returns 1 if the predicted sequence matches the gold one, and 0 otherwise. We initialize $F(\cdot)$ to 0 ($F_0(\cdot) = 0$).

We can approximate the negative functional gradient $-g_m(\cdot)$ with a regression tree model $h_m(\cdot)$ by fitting the training data $\{\phi(\mathbf{x}^{(i)}, y_t^{(i)}, \mathbf{y}_{1:t-1}^{(i)})\}$ to the point-wise negative functional gradients (also known as residuals) $\{-g_m(\mathbf{x}^{(i)}, y_t^{(i)}, \mathbf{y}_{1:t-1}^{(i)})\}$. Then the factor scoring

function can be obtained by

$$F(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}) = \sum_{m=1}^{M} \eta_m h_m(\mathbf{x}, y_t, \mathbf{y}_{1:t-1}), \tag{7}$$

where $h_m(\mathbf{x}, y_t, \mathbf{y}_{1:t-1})$ is called a basis function. We set $\eta_m = 1$ in this work.

3 Training

Training the SGTB model requires computing the point-wise functional gradients with respect to training documents and candidate entity sequences. This is challenging, due to the exponential output structure search space. First, we are not able to enumerate all possible candidate entity sequences. Second, computing the conditional probabilities shown in Eq. 6 is intractable, as it is prohibitively expensive to compute the partition function $Z(\mathbf{x})$ in Eq. 2. Beam search can be used to address these problems. We can compute point-wise functional gradients for candidate entity sequences in the beam, and approximately compute the partition function by summing over the elements in the beam.

In this section, we present a bidirectional beam search training algorithm that always keeps the gold sequence in the beam. The algorithm is tailored to SGTB, and improves standard training methods in two aspects: (1) it reduces model variance by collecting more point-wise function gradients to train a regression tree; (2) it leverages information from both past and future to conduct better local search.

3.1 Beam search with gold path

The early update (Collins and Roark, 2004) and LaSO (Daumé III and Marcu, 2005; Xu and Fern, 2007) strategies are widely adopted with beam search for updating model parameters in previous work. Both methods keep track of the location of the gold path in the beam while decoding a training sequence. A gradient update step will be taken if the gold path falls out of the beam at a specific time step t or after the last step T. Adapting the strategies to SGTB training is straightforward. We will compute point-wise functional gradients for all candidate entity sequences after time step T or when the gold sequence falls out the beam. Both early update and LaSO are typically applied to online learning scenarios, in which model parameters are updated after passing one or a few training sequences.

SGTB training, however, fits the batch learning paradigm. In each training epoch, a SGTB model will be updated only once using the regression tree model fit on the point-wise negative functional gradients. The gradients are calculated with respect to the output sequences obtained from beam search. We propose a simple training strategy that computes and collects point-wise functional gradients at every step of a training sequence. In addition, instead of passively monitoring the gold path, we always keep the gold path in the beam to ensure that we have valid functional gradients at each time step. The new beam search training method, Beam Search with Gold path (BSG), generates much more point-wise functional gradients than early update or LaSO, which can reduce the variance of the auxiliary regression tree model. As a result, SGTB trained with BSG consistently outperforms early update or LaSO in our exploratory experiments, and it also requires fewer training epochs to converge.[2]

3.2 Bidirectional beam search

During beam search, if we consider a decision made at time step t, the joint probability $p(\mathbf{y}|\mathbf{x})$ can be factorized around t as follows:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}_{1:t-1}|\mathbf{x}) \cdot p(y_t|\mathbf{y}_{1:t-1}, \mathbf{x}) \\ \cdot p(\mathbf{y}_{t+1:T}|y_t, \mathbf{y}_{1:t-1}, \mathbf{x}). \end{aligned} \tag{8}$$

Traditional beam search performs inference in a unidirectional (left-to-right) fashion. Since the beam search at time step t considers only the beam sequences that were committed to so far, $\{\mathbf{y}_{1:t-1}\}$, it effectively approximates the above probability by assuming that all futures are equally likely, i.e. $p(\mathbf{y}_{t+1:T}|y_t, \mathbf{y}_{1:t-1}, \mathbf{x})$ is uniform. Therefore, at any given time, there is no information from the future when incorporating the global structure.

In this work, we adopt a Bidirectional Beam Search (BiBS) methodology that incorporates multiple beams to take future information into account (Sun et al., 2017). It makes two simplifying assumptions that better approximate the joint probability above while remaining tractable: (1) future predictions are independent of past predictions given y_t; (2) $p(y_t)$ is uniform. These yield the following approximation:

$$\begin{aligned} p(\mathbf{y}_{t+1:T}|y_t, \mathbf{y}_{1:t-1}, \mathbf{x}) = p(\mathbf{y}_{t+1:T}|y_t, \mathbf{x}) \\ \propto p(y_t|\mathbf{y}_{t+1:T}, \mathbf{x}) \cdot p(\mathbf{y}_{t+1:T}|\mathbf{x}). \end{aligned} \tag{9}$$

[2] Early update and LaSO perform similarly, thus we only report results for early update in § 5.

Substituting this back into Eq. 8 therefore yields:

$$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{y}_{1:t-1}|\mathbf{x}) \cdot p(y_t|\mathbf{y}_{1:t-1}, \mathbf{x}) \cdot p(y_t|\mathbf{y}_{t+1:t}, \mathbf{x}) \cdot p(\mathbf{y}_{t+1:T}|\mathbf{x}), \quad (10)$$

which decomposes into multiplication of a forward probability and a backward probability. In (Sun et al., 2017), these are retrieved from forward and backward recurrent networks, whereas in our work we use the joint scores (log probabilities shown in Eq. 1) computed for partial sequences from forward and backward beams.

Algorithm 1: Bidirectional Beam Search with Gold path (BiBSG)

Input : input document $\mathbf{x}$, candidate sequences $\{\mathbf{y}\}$, joint scoring function $S(\mathbf{x}, \mathbf{y}_{t_1:t_2})$
Output: beam sequence set C
$C \leftarrow \emptyset$
while *not converged* **do**
  // forward beam search
  for $t = 1, \cdots, T$ **do**
    $C^{(F)} \leftarrow \text{top-B}_{\mathbf{y}_{1:t}}[S(\mathbf{x}, \mathbf{y}_{1:t}) + S(\mathbf{x}, \mathbf{y}_{T:t})]$
    // add gold subsequence
    $C^{(F)} \leftarrow C^{(F)} \cup \{\mathbf{y}^*_{1:t}\}$
    $C \leftarrow C \cup C^{(F)}$
  end
  // backward beam search
  for $t = T, \cdots, 1$ **do**
    $C^{(B)} \leftarrow \text{top-B}_{\mathbf{y}_{T:t}}[S(\mathbf{x}, \mathbf{y}_{T:t}) + S(\mathbf{x}, \mathbf{y}_{1:t})]$
    // add gold subsequence
    $C^{(B)} \leftarrow C^{(B)} \cup \{\mathbf{y}^*_{T:t}\}$
    $C \leftarrow C \cup C^{(B)}$
  end
end

The full inference algorithm, Bidirectional Beam Search with Gold path (BiBSG), is presented in Alg. 1. When performing the forward pass to update the forward beam, forward joint scores, $S(\mathbf{x}, \mathbf{y}_{1:t})$, are computed with respect to current forward beam, and backward joint scores, $S(\mathbf{x}, \mathbf{y}_{T:t})$, are computed with respect to previous backward beam. A similar procedure is used for the backward pass. The search converges very fast, and we use two rounds of bidirectional search as a good approximation. Finally, SGTB-BiBSG compares the conditional probabilities $p(\mathbf{y}^{(\cdot)}|\mathbf{x})$ of the best scoring output sequences $\mathbf{y}^{(\text{F})}$ and $\mathbf{y}^{(\text{B})}$ obtained from the forward and backward beams. The final prediction is the sequence with the higher conditional probability score.

4 Implementation

We provide implementation details of our SGTB systems, including entity candidate generation, adopted local and global features, and some efforts to make training and inference faster.

4.1 Candidate selection

We use a mention prior $\hat{p}(y|x)$ to select entity candidates for a mention x. Following Ganea and Hofmann (2017), the prior is computed by averaging mention prior probabilities built from mention-entity hyperlink statistics from Wikipedia[3] and a large Web corpus (Spitkovsky and Chang, 2012). Given a mention, we select the top 30 entity candidates according to $\hat{p}(y|x)$.

We also use a simple heuristic proposed by Ganea and Hofmann (2017) to improve candidate selection for persons: for a mention x, if there are mentions of persons that contain x as a continuous subsequence of words, then we consider the candidate set obtained from the longest mention for the mention x.

4.2 Features

The feature function $\phi(\mathbf{x}, y_t, \mathbf{y_{1:t-1}})$ can be decomposed into the summation of a local feature function $\phi_L(\mathbf{x}, y_t)$ and a global feature function $\phi_G(y_t, \mathbf{y}_{1:t-1})$.

Local features We consider standard local features that have been used in prior work, including mention priors $p(y|x)$ obtained from different resources; entity popularity features based on Wikipedia page view count statistics;[4] named entity recognition (NER) type features given by an in-house NER system trained on the CoNLL 2003 NER data (Tjong Kim Sang and De Meulder, 2003); entity type features based on Freebase type information; and three textual similarity features proposed by Yamada et al. (2016).[5]

Global features Three features are utilized to characterize entity-entity relationships: entity-entity co-occurrence counts obtained from Wikipedia, and two cosine similarity scores between entity vectors based on entity embeddings from (Ganea and Hofmann, 2017) and Freebase entity embeddings released by Google[6]

[3] We use a Wikipedia snapshot as of Feb. 2017.

[4] We obtain the statistics of Feb. 2017 and Dec. 2011 from https://dumps.wikimedia.org/other/pagecounts-ez/merged/ .

[5] We obtain embeddings jointly trained for words and entities from (Ganea and Hofmann, 2017).

[6] https://code.google.com/archive/p/word2vec/

respectively. We denote the entity-entity features between entities y_t and $y_{t'}$ as $\phi_E(y_t, y_{t'})$.

At step t of a training sequence, we quantify the coherence of y_t with respect to previous decisions $\mathbf{y}_{1:y-1}$ by first extracting entity-entity features between y_t and $y_{t'}$ where $1 \leq t' \leq t-1$, and then aggregating the information to have a global feature vector $\phi_G(y_t, \mathbf{y}_{1:t-1})$ of a fixed length:

$$\phi_G(y_t, \mathbf{y}_{1:t-1}) = \sum_{t'=1}^{t-1} \frac{\phi_E(y_t, y_{t'})}{t-1} \oplus \max_{t'=1}^{t-1} \phi_E(y_t, y_{t'}),$$

where $\oplus$ denotes concatenation of vectors.

4.3 Efficiency

Global models are powerful and effective, but often at a cost of efficiency. We discuss ways to speed up training and inference for SGTB models.

Many of the adopted features such as mention priors and entity-entity co-occurrences can be extracted once and retrieved later with just a hash map lookup. The most expensive features are the cosine similarity features based on word and entity embeddings. By normalizing the embeddings to have a unit norm, we can obtain the similarity features using dot products. We find this simple preprocessing makes feature extraction faster by two orders of magnitude.

SGTB training can be easily parallelized, as the computation of functional gradients are independent for different documents. During each training iteration, we randomly split training documents into different partitions, and then calculate the point-wise functional gradients for documents of different partitions in parallel.

5 Experiments

In this section, we evaluate SGTB on some of the most popular datasets for ED. After describing the experimental setup, we compare SGTB with previous state-of-the-art (SOTA) ED systems and present our main findings in § 5.3.

5.1 Data

We use six publicly available datasets to validate the effectiveness of SGTB. AIDA-CoNLL (Hoffart et al., 2011) is a widely adopted dataset for ED based on the CoNLL 2003 NER dataset (Tjong Kim Sang and De Meulder, 2003). It is further split into training (AIDA-train), development (AIDA-dev), and test (AIDA-test) sets.[7] AQUAINT (Milne and Witten, 2008), MSNBC (Cucerzan, 2007), and ACE (Ratinov et al., 2011) are three datasets for Wikification, which also contain Wikipedia concepts beyond named entities. These datasets were recently cleaned and updated by Guo and Barbosa (2016). WIKI and CWEB are automatically annotated datasets built from the ClueWeb and Wikipedia corpora by Guo and Barbosa (2016). The statistics of these datasets are available in Table 1.

Dataset	# mention	# doc	# mention per doc
AIDA-train	18,448	946	19.5
AIDA-dev	4,791	216	22.1
AIDA-test	4,485	231	19.4
AQUAINT	727	50	14.5
MSNBC	656	20	32.8
ACE	257	36	7.1
CWEB	11,154	320	34.8
WIKI	6,821	320	21.3

Table 1: Statistics of the ED datasets used in this work.

5.2 Experimental settings

Following previous work (Guo and Barbosa, 2016; Ganea and Hofmann, 2017), we evaluate our models on both *in-domain* and *cross-domain* testing settings. In particular, we train our models on AIDA-train set, tune hyperparameters on AIDA-dev set, and test on AIDA-test set (in-domain testing) and all other datasets (cross-domain testing). We follow prior work and report in-KB accuracies for AIDA-test and Bag-of-Title (BoT) F1 scores for the other test sets.

Two AIDA-CoNLL specific resources have been widely used in previous work. In order to have fair comparisons with these works, we also adopt them only for the AIDA datasets. First, we use a mention prior obtained from aliases to candidate entities released by Hoffart et al. (2011) along with the two priors described in § 4.1. Second, we also experiment with PPRforNED, an entity candidate selection system released by Pershina et al. (2015). It is unclear how candidates were pruned, but the entity candidates generated by this system have high recall and low ambiguity, and they contribute to some of the best results reported for AIDA-test (Yamada et al., 2016; Sil et al., 2018).

[7] AIDA-dev and AIDA-test are also referred as AIDA-a and AIDA-b datasets in previous work.

Competitive systems We implement four competitive ED systems, and three of them are based on variants of our proposed SGTB algorithm.[8] *Gradient tree boosting* is a local model that employs only local features to make independent decisions for every entity mention. Note that our local model is different from that presented by Yamada et al. (2016), where they treat ED as binary classification for each mention-entity pair. *SGTB-BS* is a Structured Gradient Tree Boosting model trained with Beam Search with early update strategy. *SGTB-BSG* uses Beam Search with Gold path training strategy presented in § 3.1. Finally, *SGTB-BiBSG* exploits Bidirectional Beam Search with Gold path to leverage information from both past and future for better local search.

In addition, we compare against best published results on all the datasets. To ensure fair comparisons, we group results according to candidate selection system that different ED systems adopted.

Parameter tuning We tune all the hyperparameters on the AIDA-dev set. We use recommended hyperparameter values from scikit-learn to train regression trees, except for the maximum depth of the tree, which we choose from $\{3, 5, 8\}$. After a set of preliminary experiments, we select the beam size from $\{3, 4, 5, 6\}$. The best values for the two hyperparameters are 3 and 4 respectively. As mentioned in § 2, the learning rate is set to 1. We train SGTB for at most 500 epochs (i.e., fit at most 500 regression trees). During training, we check the performance on the development set every 25 epochs to perform early stopping. Training takes 3 hours for SGTB-BS and SGTB-BSG, and takes 9 hours for SGTB-BiBSG on 16 threads.

5.3 Results

In-domain results In-domain evaluation results are presented in Table 2. As shown, SGTB achieves much better performance than all previously published results. Specifically, SGTB-BiBSG outperforms the previous SOTA system (Ganea and Hofmann, 2017) by 0.8% accuracy, and improves upon the best published results when employing the PPRforNED candidate selection system by 1.9% accuracy. Global information is clearly useful, as it helps to boost the performance by 2-4 points of accuracy, depending on the candidate generation system. In terms of beam search training strategies, BiBSG consistently outperforms BSG and beam search with early update. By employing more point-wise functional gradients to train the regression trees and leveraging global information from both past and future to carry on local search, BiBSG is able to find better global solutions than alternative training strategies.

[8] Our implementations are based on the scikit-learn package (Pedregosa et al., 2011).

System	PPRforNED	In-KB acc.
Published results		
Lazic et al. (2015)		86.4
Huang et al. (2015)		86.6
Chisholm and Hachey (2015)		88.7
Ganea et al. (2016)		87.6
Guo and Barbosa (2016)		89.0
Globerson et al. (2016)		91.0
Yamada et al. (2016)		91.5
Ganea and Hofmann (2017)		92.2
Our implementations		
Gradient tree boosting		88.4
SGTB-BS		91.7
SGTB-BSG		92.4
SGTB-BiBSG		**93.0**
Published results		
Pershina et al. (2015)	✓	91.8
Yamada et al. (2016)	✓	93.1
Sil et al. (2018)	✓	94.0
Our implementations		
Gradient tree boosting	✓	93.1
SGTB-BS	✓	95.1
SGTB-BSG	✓	95.5
SGTB-BiBSG	✓	**95.9**

Table 2: In-domain evaluation: in-KB accuracy results on the AIDA-test set. Checked PPRforNED indicates that the system uses PPRforNED (Pershina et al., 2015) to select candidate entities.The best results are in **bold**.

Cross-domain results As presented in Table 3, cross-domain experimental results are a little more mixed. SGTB-BS and SGTB-BSG perform quite competitively compared with SGTB-BiBSG. In a cross-domain evaluation setting, the test data is drawn from a different distribution as the training data. Therefore, less expressive models may be preferred as they may learn more abstract representations that will generalize better to out-of-domain data. Nevertheless, our SGTB models achieve better performance than best published results on three of the five popular ED datasets. Specifically, SGTB-BS outperforms the prior SOTA system by absolute 4% F1 on the CWEB dataset, and SGTB-BiBSG performs consistently well across different datasets.

System	AQUAINT	MSNBC	ACE	CWEB	WIKI
Published results					
Fang et al. (2016)	88.8	81.2	85.3	-	-
Ganea et al. (2016)	89.2	91.0	88.7	-	-
Milne and Witten (2008)	85.0	78.0	81.0	64.1	81.7
Hoffart et al. (2011)	56.0	79.0	80.0	58.6	63.0
Ratinov et al. (2011)	83.0	75.0	82.0	56.2	67.2
Cheng and Roth (2013)	90.0	90.0	86.0	67.5	73.4
Guo and Barbosa (2016)	87.0	92.0	88.0	77.0	**84.5**
Ganea and Hofmann (2017)	88.5	**93.7**	88.5	77.9	77.5
Our implementations					
Gradient tree boosting	90.3	91.1	**89.2**	78.8	75.0
SGTB-BS	**90.5**	92.4	88.9	81.7	76.4
SGTB-BSG	89.4	92.5	88.6	81.7	78.4
SGTB-BiBSG	89.9	92.6	88.5	**81.8**	79.2

Table 3: Cross-domain evaluation: Bag-of-Title (BoT) F1 results on ED datasets. The best results are in **bold**.

6 Related work

Entity disambiguation Most ED systems consist of a local component that models relatedness between a mention and a candidate entity, as well as a global component that produces coherent entity assignments for all mentions within a document. Recent research has largely focused on joint resolution of entities, which is usually performed by maximizing the global topical coherence between entities. As discussed above, directly optimizing the coherence objective is computationally intractable, and several heuristics and approximations have been proposed to address the problem. Hoffart et al. (2011) use an iterative heuristic to remove unpromising mention-entity edges. Yamada et al. (2016) employ a two-stage approach, in which global information is incorporated in the second stage based on local decisions from the first stage. Approximate inference techniques have been widely adopted for ED. Cheng and Roth (2013) use an integer linear program (ILP) solver. Belief propagation (BP) and its variant loopy belief propagation (LBP) have been used by Ganea et al. (2016) and Ganea and Hofmann (2017) respectively. We employ another standard approximate inference algorithm, beam search, in this work. To make beam search a better fit for SGTB training, we propose BiBSG that improves beam search training on stability and effectiveness.

Structured gradient tree boosting Gradient tree boosting has been used in some of the most accurate systems for a variety of classification and regression problems (Babenko et al., 2011; Wu et al., 2010; Yamada et al., 2016). However, gradient tree boosting is seldom studied in the context of structured learning, with only a few exceptions. Dietterich et al. (2004) propose TreeCRF that replaces the linear scoring function of a CRF with a scoring function given by a gradient tree boosting model. TreeCRF achieves comparable or better results than CRF on some linear chain structured prediction problems. Bagnell et al. (2007) extend the Maximum Margin Planning (MMP; Ratliff et al., 2006) algorithm to structured prediction problems by learning new features using gradient boosting machines. Yang and Chang (2015) present a general SGTB framework that is flexible in the choice of loss functions and specific structures. They also apply SGTB to the task of tweet entity linking with a special non-overlapping structure. By decomposing the structures into local substructures, exact inference is tractable in all the aforementioned works. Our work shows that we can train SGTB models efficiently and effectively even with approximate inference. This extends the utility of SGTB models to a wider range of interesting structured prediction problems.

7 Conclusion and future work

In this paper, we present a structured gradient tree boosting model for entity disambiguation. Entity coherence modeling is challenging, as exact inference is prohibitively expensive due to the pairwise entity relatedness terms in the objective function. We propose an approximate inference algorithm, BiBSG, that is designed specifically for SGTB to solve this problem. Experiments on benchmark ED datasets suggest that the expressive SGTB models are extremely good at dealing with the task of ED. SGTB significantly outperforms all previous systems on the AIDA-CoNLL dataset,

and it also achieves SOTA results on many other ED datasets even in the cross-domain evaluation setting. SGTB is a family of structured learning algorithms that can be potentially applied to other core NLP tasks. In the future, we would like to investigate the effectiveness of SGTB on other information extraction tasks, such as relation extraction and coreference resolution.

8 Acknowledgments

We thank Prabhanjan Kambadur and other people in the Bloomberg AI team for their valuable comments on earlier version of this paper. We also thank the NAACL reviewers for their helpful feedback. This work also benefitted from discussions with Mark Dredze and Karl Stratos.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the Association for Computational Linguistics (ACL)*.

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* .

JA Bagnell, Joel Chestnutt, David M Bradley, and Nathan D Ratliff. 2007. Boosting structured prediction for imitation learning. In *Neural Information Processing Systems (NIPS)*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Andrew Chisholm and Ben Hachey. 2015. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics* 3.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Hal Daumé III and Daniel Marcu. 2005. Learning as search optimization: Approximate large margin methods for structured prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*.

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)*.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14(2).

Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. 2016. Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. 2016. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the International Conference on World Wide Web (WWW)*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. In *Proceedings of the Association for Computational Linguistics (ACL)*.

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)*.

Zhaochen Guo and Denilson Barbosa. 2016. Robust named entity disambiguation with random walks. *Semantic Web* .

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8).

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Hongzhao Huang, Larry Heck, and Heng Ji. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678* .

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the Association for Computational Linguistics (ACL)*.

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)*.

Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2015. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics* .

David Milne and Ian H Witten. 2008. Learning to link with wikipedia. In *Proceedings of the ACM conference on Information and knowledge management (CIKM)*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* .

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. 2006. Maximum margin planning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases* .

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Noah A Smith and Mark Johnson. 2007. Weighted and probabilistic context-free grammars are equally expressive. *Computational Linguistics* .

Valentin I Spitkovsky and Angel X Chang. 2012. A cross-lingual dictionary for english wikipedia concepts. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Qing Sun, Stefan Lee, and Dhruv Batra. 2017. Bidirectional beam search: Forward-backward inference in neural sequence models for fill-in-the-blank image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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

Yuehua Xu and Alan Fern. 2007. On learning linear ranking functions for beam search. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Yi Yang and Ming-Wei Chang. 2015. S-mart: Novel tree-based structured learning algorithms applied to tweet entity linking. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the Association for Computational Linguistics (ACL)*.