AN ABSTRACT OF THE THESIS OF

Jun Xie for the degree of Master of Science in Computer Science presented on March 4, 2014.

Title: Learning Greedy Policies for the Easy-First Framework

Abstract approved: _____

Xiaoli Z. Fern

Easy-first, a search-based structured prediction approach, has been applied to many NLP tasks including dependency parsing and coreference resolution. This approach employs a learned greedy policy (action scoring function) to make easy decisions first, which constrains the remaining decisions and makes them easier. This thesis studies the problem of learning greedy policies to ensure the success of the Easy-first approach. We propose a novel and principled online learning algorithm that solves an optimization problem to update the weights whenever a mistake happens so that 1) the greedy policy chooses a correct action, and 2) the change to the function is minimized (to avoid over-fitting). The proposed objective is non-convex and optimized via an efficient Concave-Convex Procedure (CCCP). We compare the proposed approach with existing learning approaches on both within-document coreference and cross-document joint entity and event coreference tasks. Results demonstrate that the proposed approach performs better than existing training regimes for Easy-first and is less susceptible to overfitting.

Learning Greedy Policies for the Easy-First Framework

by

Jun Xie

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 4, 2014
Commencement June 2014

Master of Science thesis of Jun Xie presented on March 4, 2014.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Jun Xie, Author

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Xiaoli Z. Fern for the continuous support of my graduate study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my graduate study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Thomas G. Dietterich and Prof. Prasad Tadepalli, for their encouragement, insightful comments, and hard questions.

My sincere thanks also goes to Janardhan Rao Doppa for his help and encouragement.

I thank my fellow students in the project of Deep Reading and Learning: Chao Ma, John Walker Orr, Prashanth Mannem, and Shahed Sorower, for the stimulating discussions.

Last but not the least, I would like to thank my wife Rui Qin for her understanding and love during the past years.

TABLE OF CONTENTS

Page

1 Introduction 1

2 Related Work 3

2.1 The Easy-first Framework . . . 3

2.2 Within-document Entity Coreference Resolution . . . 4

2.3 Joint Entity and Event Coreference Resolution across Documents . . . 6

3 Proposed Greedy Policy Learning Algorithm 8

3.1 Easy-first Framework and Baseline . . . 8

3.1.1 Easy-first: inference and training . . . 8

3.1.2 A baseline update strategy . . . 10

3.2 Best Good Violated Bad . . . 11

4 Experimental Evaluation 15

4.1 Evaluation Metrics . . . 15

4.2 Entity coreference resolution within documents . . . 16

4.3 Joint entity and event coreference resolution across documents . . . 17

4.4 Summary of results . . . 20

4.5 Discussion . . . 21

5 Conclusions and Future Work 23

Bibliography 23

LIST OF FIGURES

Figure Page

3.1 A visual illustration of the difference between two update rules. The circles represent actions ordered in increasing score. BGBB focuses on the best good and best bad actions where BGVB considers the best good action and all the bad actions that scored higher than it. 14

LIST OF TABLES

Table Page

4.1 Performance of our method (BGVB) and baseline (BGBB) on the ACE2004 corpus. 16

4.2 Performance of our method (BGVB) and baseline (BGBB) on the MUC6 corpus. 17

4.3 Cross-document joint entity and event coreference resolution (on-trajectory training) on the EECB corpus using BGVB approach. Detailed information about the first column: Set. (Setting), Enti. (Entity) and Even. (Event). The name of the last column is CoNLL F1. The following tables follow the same convention. . 17

4.4 Cross-document joint entity and event coreference resolution (on-trajectory training) on the EECB corpus using BGBB approach. 18

4.5 Cross-document joint entity and event coreference resolution (off-trajectory training) on the EECB corpus using BGVB approach. 18

4.6 Cross-document joint entity and event coreference resolution (off-trajectory training) on the EECB corpus using BGBB approach. 19

4.7 Cross-document joint entity and event coreference resolution (off-trajectory training) on the EECB corpus using Stanford system. 19

4.8 Corpus Statistics. 20

4.9 Training statistics on MUC-6 and ACE04 corpora 21

4.10 Global performance of the learned weights. 21

LIST OF ALGORITHMS

Algorithm		Page
1	Easy-first Inference	9
2	Online training for Easy-first (single iteration)	10
3	The CCCP procedure to solve Equation 3.4	13

Chapter 1: Introduction

Easy-first is a general search-based structured prediction framework that has been successfully applied to a variety of natural language processing (NLP) tasks including POS tagging [36], dependency parsing [15] and coreference resolution [39]. In this framework, the output is constructed incrementally by making the easiest (most confident) decision at each decision step to gather more evidence for making hard decisions later. A decision might be selecting an edge to be added between a particular pair of nodes for dependency parsing or selecting two clusters to be merged for co-reference resolution. Consider the following example from the EECB corpus for the problem of joint entity and event coreference resolution across documents [24].

(a) **Hugh Jackman** plays a furry comic-book hero.

(b) **The Australian actor** is playing a super hero.

The coreference resolution decision for the two verb mentions "plays" and "is playing" is easy, because they share the same lemma "play". In contrast, the coreference decision for the two noun mentions "Hugh Jackman" and "The Australian actor" is hard based on the lexical, syntactic, semantic and discourse constraints or features [19, 28]. In the Easy-first framework, the easiest decision is made early. Once we establish the fact that the two verbs are coreferent, we have stronger evidence suggesting that the two noun mentions are coreferent because they serve the same semantic role to the same verb cluster.

The Easy-first approach is a greedy search process that selects the easiest decision (i.e., a decision with the highest score according to a learned policy) at each decision-making step. Because it is applied greedily, the learned policy plays a critical role in the effectiveness of this approach. The focus of this thesis is to study principled ways of learning policies to ensure the success of the Easy-first framework. In particular, we propose a novel online learning algorithm that learns a linear policy for the Easy-first approach. The contributions of this work are as follows:

- We formulate greedy policy learning as optimizing a non-convex objective consisting of two parts. The first part employs hinge loss to ensure that the learned policy ranks at least one good action higher than all the bad actions. The second part regularizes the weight vector to avoid overly-aggressive updates and over-fitting.

- We develop an efficient Concave-Convex Procedure (CCCP) procedure to optimize the proposed non-convex objective function, which can be easily expressed as the difference of two convex functions.
- We evaluate our approach in two NLP domains: within-document entity coreference resolution and cross-document joint entity and event coreference resolution. Our results demonstrate that the proposed approach performs better than the baseline training approaches for the Easy-first framework and is less prone to overfitting.

The thesis is organized as follows.

- The related work is reviewed in Chapter 2. We provide related work for three topics, including the Easy-first framework, within-document entity coreference resolution, and cross-document joint entity and event coreference resolution.
- We present our proposed learning algorithm in Chapter 3. Before introducing the proposed model, we define the notation used throughout the thesis, and we also describe a baseline learning algorithm for the Easy-first approach.
- The experimental results and discussions are presented in Chapter 4. First, we introduce the metrics for evaluating the performance of co-reference resolution systems. Then we describe the experiment setup and results for two coreference resolution problems. Finally, we present our analysis and discussion on the results we got.
- We conclude the thesis in Chapter 5.

Chapter 2: Related Work

This thesis mainly focuses on applying the Easy-first framework in two NLP domains: within-document entity coreference resolution and joint entity and event coreference resolution across documents. We describe the related work for each topic below.

2.1 The Easy-first Framework

Many sequence labeling algorithms employ recurrent classifiers that take advantage of the previous labels to decide the next label [11, 8]. To overcome the deficiency of a fixed labeling order (e.g., left-to-right), some researchers have employed Reinforcement Learning (RL) to learn the ordering [27]. Such approaches have a flavor similar to the Easy-first framework in that the ordering of the decisions are determined based on some learned functions.

The Easy-first approach was informally introduced by Shen et al. [36] as a *bidirectional* labeling scheme for sequence labeling problems. More recently, Goldberg and Elhadad [15] formally introduced the Easy-first framework and applied it to dependency parsing. At the same time, Raghunathan et al. [31] introduced a multi-sieve model for co-reference resolution. The multi-pass sieve system for coreference resolution applies a set of hand-designed rules ranked according to their precision (high to low). It can be considered as an instance of the Easy-first framework, where easy decisions are made first via high-precision rules (e.g., the exact string match rule) to aggregate information for making later decisions with low-precision rules (e.g., the pronoun resolution rule). The order of decisions was determined manually for the system.

Lee et al. [24] and Stoyanov and Eisner [39] subsequently applied the same idea to coreference resolution, but aimed to learn the order of decisions instead of setting the order manually based on prior knowledge. The basic idea is to construct the (structured) output incrementally by making the easiest (most confident) decision at each step so that it constrains the remaining decisions. A learned policy is employed to select the "easiest" decision. Different learning strategies have been proposed in the literature. In particular, Lee et al. [24] learns the greedy policy using a linear regression model trained in an off-line batch fashion. In contrast, Goldberg [15] and Stoyanov [39] train linear scoring functions using the online structured perceptron where, for

each mistake, one or more perceptron updates are performed using the features of the current best-scoring good decision and best-scoring bad decision.

Easy-first is equivalent to the *LaSO* framework [9, 41] instantiated with greedy search, and it differs only in the way the policy is updated. Specifically, when a mistake happens, LaSO performs a single perceptron update between the averaged features of all good decisions and the best-scoring bad decision [41]. The training methodology employed by the Easy-first approach can be considered as an online version of the DAgger algorithm [35] for imitation learning with one difference – DAgger assumes the availability of a single optimal decision at each state (deterministic oracle), whereas Easy-first allows multiple optimal decisions at each state (non-deterministic oracle) and the training approach breaks ties based on the current policy scores. In spite of their success, existing approaches lack strong theoretical guarantees, and their convergence results rely on strong assumptions [41].

All the above approaches either employ manual rules or a heuristically-motivated training procedure to select the easiest decision at each step. Our work provides a principled approach to learn the decision-making policy. The key distinction between the existing approaches [15, 39] and ours is that instead of using a heuristic weight-update rule, we formally define an objective that captures the underlying goal of the Easy-first framework and derive the corresponding update rule by optimizing that objective.

2.2 Within-document Entity Coreference Resolution

The task of entity co-reference resolution is to group together text expressions, mostly noun phrases (NPs), which refer to the same real world object. Focusing on NPs is a way to restrict the challenging problem of coreference resolution [24]. Take an excerpt as an example below. There are three NPs, including "Turkmenistan", "the energy-rich central Asian nation" and "its", all of which refer to the same entity. Hence, this task tries to cluster the three expressions together.

> "Turkmenistan voted Sunday in a parliamentary election meant to show that the energy-rich central Asian nation was shedding its autocratic past, but Western observers said nothing had changed."

Coreference resolution research has been an important and active area in the field of Natural Language Processing since the 1960s [28]. Researchers firstly employed linguistic properties to

deal with the coreference resolution problem. The most notable theories at that time were the Focusing [16, 37] and Centering Algorithms [17, 18]. As large corpora became available and machine learning methods became more mature, there has been a lot of research aimed at resolving the problem using machine learning techniques. The reason is that machine learning methods can generalize well to unseen data. The basic machine learning approach to co-reference resolution consists of two steps: the classification step and the clustering step. The first step is to determine whether the current active mention should be linked with one of previous antecedents. The second step is to coordinate the classification decisions made during the first step to fix cases where the decisions do not satisfy the transitivity property, which should be enforced for co-reference relation [28]. There are two important types of machine learning-based coreference models, namely, the mention-pair model [38] and the entity-mention model [7]. For every single classification decision, the difference between the two models is whether the active mention is linked with one of the previous mentions or linked to one of partial clusters based on the similarity between them.

Both models fall in the realm of supervised learning. There is still a lot of ongoing research in this direction [26, 3, 32]. The first work in coreference resolution based on the idea of search is Luo's work [26]. Luo trained a maximum entropy model by finding a path with the highest probability for the gold co-reference output. Bengtson and Roth [3] investigated the importance of different features for the task of co-reference resolution based on the mention-pair model. In addition, Cluster Ranking Models [32] create a competition between all candidate antecedents for the active mention and choose the best one. In contrast to the main-stream supervised methods for co-reference resolution, Poon and Domingos [29] proposed a jointly un-supervised coreference resolution model based on Markov Logic.

In the CoNLL 2011 shared task, the fact that Stanford Multi-Sieve System [23] performed the best in several sub-tasks drew a lot of attention, because the system is purely based on linguistic rules. Those rules are modularized as "Sieves" and are ranked by the precision. Each document is scanned from the beginning of the document to the end of the document by those sieves. During this process, the mentions are clustered together based on the rules defined in the active sieve and the output of the previously-applied sieves, if any. There are two impacts to the coreference resolution research brought by this paper. The first is that it led to an increase in the amount of research exploring the "Sieve" concept. The assumption is that different mention types need different coreference models [10]. The second impact is that the tendency to combine the rule-based and machine learning methods. For example, in the CoNLL 2012 shared task, there are

several high ranking systems that apply Stanford's multi-sieve system to process the corpus first in order to get an initial clustering result, and then employ the machine learning methods to tune the parameters of their models in order to improve the performance. For example, Chen and Ng [4] adopted a hybrid approach to tackle the multi-lingual coreference resolution problem. The method combined the strengths of rule-based methods and learning-based methods.

As we mentioned earlier, Stanford's multi-Sieve System [23] determines the order of decisions manually. Recently, Stoyanov and Eisner [39] learned a greedy heuristic function to determine the order of decisions based on the Easy-first framework [15].

2.3 Joint Entity and Event Coreference Resolution across Documents

The problem of joint entity and event coreference resolution across documents resolves the event and entity mentions into their corresponding clusters simultaneously. Compared with entity coreference resolution, which is relatively easy to achieve by exploiting the syntactic and semantic properties, this problem is much harder. For event mention, verb phrase is the common referent. Take the EECB corpus [24] as an example. Here is one excerpt from the corpus:

> People said Reid's representative Jack Ketsoyan confirmed the actress's stay at the Promises.

According to the EECB annotation [24], the two verb phrase "said" and "confirmed" are event mentions. Besides verb phrases, noun phrases can also be used to represent events. Here, the word "stay" is a noun phrase and is also used to denote an event mention. In this way, this problem needs to take care of both event mention types.

Event Co-reference resolution and Entity Co-reference resolution can benefit from each other by taking advantage of the semantic relationship between them. In the example above, the entity mention, "Reid's representative Jack Ketsoyan", is the subject of the event mention, "confirmed". If two event mentions have the same subject, then there is a high chance that the two event mentions should be resolved together. This also applies to the entity mentions. Recently, there is some co-reference resolution work based on this idea. For entity co-reference resolution, Haghighi and Klein [19] used the governor of the head of noun phrases as a feature in a mention-pair model, while Rahman and Ng [32] also incorporate the semantic roles that the entity mentions serve in the semantic relationship with the verb. For event co-reference resolution, Chen and Ji [5] proposed a spectral graph clustering model to address the event co-reference

resolution problem by assuming the gold entity clusters. This allows the graph clustering model take advantage of the entity information involved in the semantic relationships. In addition, Bejan and Harabagiu [2] introduced a non-parametric Bayesian model to address the open-domain event co-reference task. This work also assumed that the gold entity clusters are available. The work mentioned above just deals with one aspect of Entity and Event co-reference resolution, instead of solving both jointly. There has been little work in the direction of solving entity and event co-reference resolution jointly [21, 20]. Most recently, Lee et al. [24] proposed an iterative joint model on Entity and Event co-reference resolution across documents. The joint model is linked based on the semantic role labeling features. When two clusters (can be event or entity) are merged together, all the mention features are regenerated to reflect the current clusters. Lee et al. [24] trained a linear regression model in an off-line batch fashion and employed the learned greedy policy to guide the search to construct the final co-reference resolution result.

Chapter 3: Proposed Greedy Policy Learning Algorithm

In this chapter, we propose a greedy policy learning algorithm in the Easy-first framework by formulating the learning problem within Easy-first as an optimization objective and applying a Concave-Convex Procedure (CCCP) to optimize the proposed objective. Before we propose our approach, we give an overall view of Easy-first framework and describe our policy learning baseline.

3.1 Easy-first Framework and Baseline

This section first formally introduces the Easy-first framework and presents a generic online training procedure. We then describe a popular online learning algorithm for Easy-first, which serves as our baseline.

3.1.1 Easy-first: inference and training

Given structured inputs $x \in \mathcal{X}$ and outputs $y \in \mathcal{Y}$, we assume a task-specific non-negative loss function L. The loss function $L(x, y', y) : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \mapsto \mathcal{R}^+$ associates a loss with labeling an input x with y' when its true output is y. There are two key elements in the Easy-first framework: 1) the *search space* $\mathcal{S}_p$, whose states correspond to partial structured outputs, and 2) an *action scoring function* f, which is used to construct the complete structured output.

The search space, $\mathcal{S}_p$ is a 2-tuple $\langle I, A \rangle$, where I is the initial state function, and $A(s)$ is a function that gives the set of allowed actions from a given state. Given the actions $A(s)$ from a specific state s, we consider any action $a \in A(s)$ that results in a state s' with $L(s') < L(s)$ as a *good action*; otherwise, it is a *bad action*. Within the Easy-first framework, it is typical to encounter states that have more than one good action. We denote the set of all good actions in state s as $G(s)$ and the set of all bad actions as $B(s)$ ($G(s) \cup B(s) = A(s)$).

The second element, the action scoring function f, evaluates all actions in $A(s)$ and guides the search to incrementally produce the structured output. Given a structured input $x \in \mathcal{X}$ and scoring function f, the Easy-first inference procedure is illustrated in Algorithm 1. The search

Algorithm 1 Easy-first Inference

1: **input** : Structured Input x, action scoring function f
2: $s \leftarrow \mathrm{I}(x)$ // get initial state of x
3: TERMINATE $\leftarrow False$
4: **while not** TERMINATE **do**
5: $\quad a_p \leftarrow \arg\max_{a \in A(s)} f(a)$
6: $\quad s \leftarrow$ Apply a_p on s
7: $\quad$ **if** Terminal(s) or $a_p = HALT$ **then**
8: $\quad\quad$ TERMINATE $\leftarrow True$
9: $\quad$ **end if**
10: **end while**
11: **output** : s

greedily traverses the search space $\mathcal{S}_p$. In any state s, the scoring function f is applied to evaluate the quality of each action $a \in A(s)$. The action with the highest score is executed. This process is repeated until a terminal state is reached (for problems with a natural notion of terminal states, e.g., dependency parsing) or a HALT action is chosen (for problems like coreference resolution where we need to learn when to stop) and the predicted output is returned. In this work, we consider linear action-scoring functions $f(a) = w \cdot \phi(a)$, where w is the weight vector and $\phi(\cdot)$ is a predefined feature function. Note that the linear function can have great flexibility in terms of the features that we could consider for characterizing actions. For instance, for coreference resolution an action (i.e., a merge between two clusters) can be described by a set of local features characterizing the pair of clusters to be merged, or by features describing the global characteristics of the state (partial solution) obtained by the merge action.

The success of the Easy-first framework hinges upon the ability to choose a good action in each decision step. Hence, the *learning goal* within the Easy-first framework is to learn a weight vector w such that the highest scoring action in each step is a *good* action. Toward this goal, a general online training procedure is described in Algorithm 2. In any given state s, we assume that there exists an oracle O that can identify $G(s)$ (line 7), the set of all *good* actions given the current state and the expected output. If the current highest scoring action a_p is a good action, there is no need to update. Otherwise, the weight is updated (lines 9-12). A ChooseAction procedure is then called to select the next action, and we transit to the next search state. This is repeated until the termination condition is met (line 15). Algorithm 2 presents the procedure for one training iteration. This is typically repeated for multiple iterations, and the updated weights

Algorithm 2 Online training for Easy-first (single iteration)

1: **input** : Structured input and output pairs $(x_i, y_i)_{i=1}^n$, parameter-vector w
2: // online training for each pair (x_i, y_i)
3: **for** input $x_i \in \mathcal{X}$, with output $y_i \in \mathcal{Y}$ **do**
4: $s \leftarrow \mathrm{I}(x_i)$
5: TERMINATE $\leftarrow False$
6: **while not** TERMINATE **do**
7: $G(s) \leftarrow O(s)$
8: $B(s) \leftarrow A(s) - G(s)$
9: $a_p \leftarrow \arg\max_{a \in A(s)} w \cdot \phi(a)$
10: **if** $a_p \in B(s)$ **then**
11: UPDATE$(w, G(s), B(s))$
12: **end if**
13: $a_c \leftarrow$ ChooseAction$(A(s))$
14: $s \leftarrow$ Apply a_c on s
15: **if** Terminal(s) or $a_c = HALT$ **then**
16: TERMINATE $= True$
17: **end if**
18: **end while**
19: **end for**
20: **output** : w

are collected along the way and averaged at the end (which reduces overfitting and improves performance).

There are two elements that need to be specified in this basic training procedure. First, how to perform the update (line 11), which is the main focus of this thesis. Second, how to choose the next action, which determines the training trajectories (line 13). Two types of approaches have been pursued in the literature for this purpose: on-trajectory training, which always chooses an action in $G(s)$ (e.g., the highest-scoring action in $G(s)$), and off-trajectory training, which always chooses the highest-scoring action based on the current scoring function even when it is a bad action. In this work, we consider both types of training trajectories.

3.1.2 A baseline update strategy

Now we introduce a popular update strategy that has been widely employed in prior Easy-first work [15, 39]. This strategy aims to update the weights so that one of the good actions will score

the highest. To achieve this, it uses a simple heuristic by focusing on the highest scoring good action, referred to as g^*, and the highest scoring bad action, referred to as b^*, and adjusting the weights to increase $f(g^*)$ and decrease $f(b^*)$. For this reason, we refer to this method as Best Good Best Bad (BGBB). The specific update rule for BGBB is given by Equation 3.1.

$$w \leftarrow w - \eta \cdot (\phi(b^*) - \phi(g^*)), \tag{3.1}$$

where η is the learning rate. This update is typically done repeatedly for a fixed number of iterations or until a good action is scored the highest. In each iteration, g^* and b^* are re-evaluated. While this heuristic update has been widely applied [15, 39], it has a number of issues. Although it updates the weight in a direction that promotes a good action and demotes a bad action, there is no guarantee that the final goal (ranking a good action above all bad actions) can actually be achieved. Very frequently, even after a large number of iterations, the updated weight still chooses a bad action. In some cases there may not exist a weight vector that ranks a good action higher than all bad actions. In such cases, the effect of the heuristic update rule is unclear, because it lacks an explicit optimization objective.

3.2 Best Good Violated Bad

Our goal is to *learn a linear scoring function f such that in any given state s a good action is scored higher than all bad action.* This goal can be captured by the following set of constraints:

$$\max_{a \in G(s)} f(a) > f(b) \quad \forall b \in B(s) \tag{3.2}$$

That is, the score of the highest scoring good action needs to exceed the score of any bad action. If we identify a weight vector w that enables f to satisfy these constraints for a given s, then Easy-first would choose a correct action in state s. Because it is not always possible to find a w that satisfies all the constraints, we introduce the following average hinge loss function to capture them as soft constraints.

$$L_h(w) = \frac{1}{|B(s)|} \sum_{b \in B(s)} [1 - \max_{a \in G(s)} w \cdot \phi(a) + w \cdot \phi(b)]_+ \tag{3.3}$$

where $[\cdot]_+ = \max(0, \cdot)$, $B(s)$ and $G(s)$ denote the set of bad and good actions in state s, and $\phi(\cdot)$ returns the feature vector representing the input action.

Additionally, the weights should be only updated as much as necessary to satisfy these constraints. This is inspired by passive-aggressive Perceptron training [6] and helps avoid overly aggressive updates that can lead to over-fitting. Combining the two parts, our objective can be described as follows (w_0 represents the current weight vector prior to update):

$$\arg\min_{w} \lambda\|w - w_0\|^2 + \frac{1}{|B(s)|}\sum_{b\in B(s)}[1 - \max_{a\in G(s)} w\cdot\phi(a) + w\cdot\phi(b)]_+ \tag{3.4}$$

where λ trades-off the two aspects of the objective.

While the hinge loss is a convex function, the negative max inside the hinge loss makes the objective non-convex. Fortunately it is straightforward to show that the objective can be expressed as a difference of convex functions, making it possible to apply a Concave-Convex Procedure (CCCP) [42] to find a local optimal solution. We describe our CCCP algorithm in Algorithm 3.

CCCP works in iterations. In each iteration, we convexify the objective by replacing $max_{a\in G(s)} w\cdot\phi(a)$ with $w\cdot\phi(g^*)$, where g^* is the current best good action:

$$\arg\min_{w} \lambda\|w - w_0\|^2 + \frac{1}{|B(s)|}\sum_{b\in B(s)}[1 - w\cdot\phi(g^*) + w\cdot\phi(b)]_+ \tag{3.5}$$

This convex objective is then optimized via gradient descent (line 8). The optimal weight vector is then used to identify the g^* in order to convexify the objective for the next iteration. This repeats until convergence (lines 10-12) or for a fixed number of iterations (T).

We omit the details of the gradient descent subroutine for solving Equation 3.5. However, it is worth noting that each step of the gradient descent procedure corresponds nicely to a single update step of the Best Good Best Bad approach (Equation 3.1). In particular, each gradient descent step performs the following:

$$w \leftarrow w - \eta\left[\lambda(w - w_0) + \frac{1}{|B(s)|}\sum_{a\in V}\phi(a) - \phi(g^*)\right], \tag{3.6}$$

where η is the learning rate and V is a subset of $B(s)$ that contains all the bad actions that scored higher than g^*.

Comparing to Equation 3.1, we note two key differences. First, our update rule does not solely focus on the best bad action. Instead, it tries to suppress all the bad actions that incur

Algorithm 3 The CCCP procedure to solve Equation 3.4

input : current parameter-vector w_0, convergence threshold ϵ, λ, good actions G, bad actions B, max number of iterations T
output : w
$w \leftarrow w_0$, $i \leftarrow 0$, convergence $\leftarrow$ false
$pObj \leftarrow -\infty$
while $i \leq T$ and !convergence **do**
  $i \leftarrow i + 1$
  $g^* \leftarrow \arg\max_{a \in G} w \cdot \phi(a)$
  $w \leftarrow$ solve Equation 3.5 via gradient descent
  $cObj \leftarrow$ evaluate the objective in Equation 3.4
  if $cObj - pObj \leq \epsilon$ **then**
    convergence $\leftarrow true$
  end if
  $pObj \leftarrow cObj$
end while

constraint violations. Hence, we refer to our update rule as Best Good Violated Bad (BGVB). See Figure 3.1 for a visual demonstration of the difference between the two update rules. We argue that by considering all violated bad actions at once, we avoid the jumpy behavior of BGBB from one iteration to the next and increase learning stability.

The second key difference is that our update rule has the added flexibility for explicit control of aggressiveness and overfitting in updates. By tuning parameter λ, we can achieve a trade-off between aggressively satisfying the given constraints and conservatively staying close to the current weight.

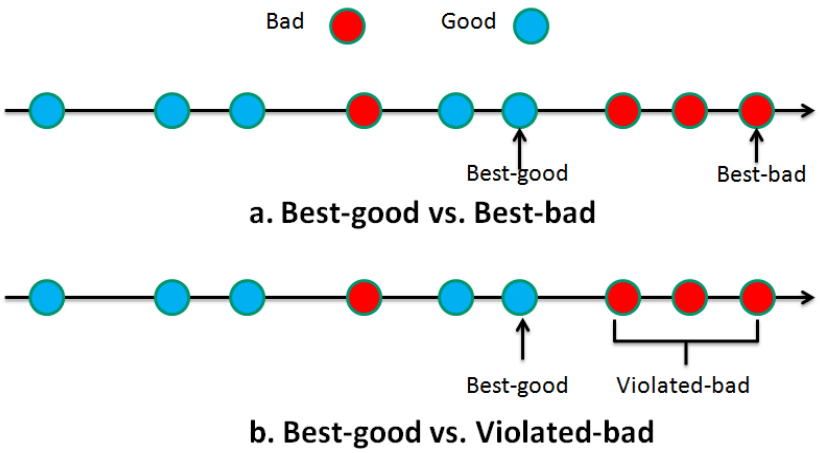

Figure 3.1: A visual illustration of the difference between two update rules. The circles represent actions ordered in increasing score. BGBB focuses on the best good and best bad actions where BGVB considers the best good action and all the bad actions that scored higher than it.

Chapter 4: Experimental Evaluation

We evaluate our approach on two NLP problem domains: within-document entity coreference resolution and joint entity and event coreference resolution across documents. Instead of just dealing with noun phrases, which is the main focus of entity co-reference resolution problem, joint entity and event co-reference resolution problem also handles verbs. To avoid the noise introduced by the mention extraction component, we conduct our evaluation using only the manually-extracted "gold mentions" for both problem domains.

4.1 Evaluation Metrics

There are several popular evaluation metrics for coreference resolution and they evaluate different aspects of the problem. In our evaluation, we consider all evaluation metrics that are employed in CoNLL Shared-task 2011: MUC, B^3, CEAF, BLANC, and CoNLL F1. Below, we briefly describe these metrics.

- **MUC**: It measures the minimal permutations that are required for the predicted (gold) output to get the gold (predicted) output [40].
- **$\mathbf{B^3}$**: There are two varinats of the B^3 metric: a) Entity-based, and b) Mention-based. The CoNLL shared-task 2011 uses Mention-based B^3 metric. Given a specific mention, it measures the overlapping mentions between the gold output and the predicted output [1].
- **CEAF**: We employ entity-based CEAF metric, called CEAF-ϕ_4. Given a bipartite graph with one-to-one alignment between the gold output and the predicted output, it measures the agreement between them [25].
- **BLANC**: This metric deals with singleton clusters in a principled manner by rewarding the correct clusters according to the number of mentions in them [34]. It examines both coreference and non-coreference links based on the Rand Index [33].
- **CoNLL F1**: This is a summary metric, which is defined as the average of MUC, B^3, and CEAF-ϕ_4 scores [30].

		MUC			B^3			CEAF-ϕ_4			
	Approach	R	P	F1	R	P	F1	R	P	F1	CoNLL F1
On-trajectory	BGVB	67.17	83.48	74.44	73.70	90.17	81.11	84.64	70.50	76.38	77.31
	BGBB	70.33	78.51	74.19	75.65	84.49	79.82	80.37	71.88	75.41	76.47
Off-trajectory	BGVB	67.76	83.17	74.68	74.38	89.26	81.14	84.44	71.12	76.73	77.52
	BGBB	67.07	81.21	73.46	74.21	88.54	80.75	83.83	70.95	76.37	76.86

Table 4.1: Performance of our method (BGVB) and baseline (BGBB) on the ACE2004 corpus.

4.2 Entity coreference resolution within documents

We first consider the problem of entity coreference resolution within documents, which groups noun phrase mentions into clusters corresponding to entities. Within-document entity coreference resolution has been widely studied, and there exist many successful systems, including the Easy-first system [39].

Data. For this problem, we conduct experiments on two widely-used entity coreference resolution corpora.

- **MUC-6:** There are 255 documents in total with 13,801 gold mentions. We use the official 30 documents for testing. Of the remaining 225 documents, we select 195 documents randomly for training, and use the remaining 30 documents as the validation set.

- **ACE04:** We employ the same training/testing partition as ACE2004-CULOTTA-TEST [7, 3]. There are 443 documents in total with 27,667 gold mentions. Among those documents, 268 documents are used for training, 68 documents for validating, and 107 documents for testing.

Experimental setup. We implemented our learning algorithm on top of the Easy-first coreference system [39]. We employ the same set of features as described by [39], which includes cluster features capturing cluster-level global agreement and mention-pair features capturing local configurations signifying coreferences. We also follow the same protocol for handling the "HALT" action (which serves as the terminal state when selected) as [39].

Our experiments compare our update strategy (BGVB) with the Best Good Best Best (BGBB) strategy of [39]. We use the same experimental setting as described by [39] except a few small differences. To ensure fair comparison of the two update rules, we initialize both methods with the zero weight vector. Another difference is that we employ five-fold cross-validation for parameter tuning for both methods. For BGBB, we tune the learning rate (η) and the number of

		MUC			B^3			CEAF-ϕ_4			
	Approach	R	P	F1	R	P	F1	R	P	F1	CoNLL F1
On-trajectory	BGVB	74.97	86.30	80.25	67.79	81.99	74.22	72.60	50.32	58.89	71.12
	BGBB	78.45	83.63	80.48	76.31	68.79	72.35	65.19	58.35	60.64	71.16
Off-trajectory	BGVB	75.03	88.57	81.22	69.78	84.73	76.53	74.16	50.14	59.25	72.33
	BGBB	71.22	91.84	80.22	65.59	90.74	76.14	75.04	42.83	54.20	70.19

Table 4.2: Performance of our method (BGVB) and baseline (BGBB) on the MUC6 corpus.

	MUC			B^3			CEAF-ϕ_4			BLANC			
Set.	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1
Enti.	74.26	83.92	78.79	59.27	80.96	68.43	53.38	37.98	44.38	72.06	83.83	76.69	63.87
Even.	66.76	73.20	69.83	51.67	79.11	62.51	41.47	34.60	37.72	69.76	84.98	75.22	56.69
Both	72.09	87.10	78.89	57.06	84.32	68.06	61.4	36.32	45.64	71.54	86.29	77.02	64.20

Table 4.3: Cross-document joint entity and event coreference resolution (on-trajectory training) on the EECB corpus using BGVB approach. Detailed information about the first column: Set. (Setting), Enti. (Entity) and Even. (Event). The name of the last column is CoNLL F1. The following tables follow the same convention.

repeated perceptron updates (k) for each mistake step. For BGVB, we tune the λ parameter for Equation 3.4, the number of gradient descent steps (t), and the number of CCCP iterations (T). Note that for gradient descent, our method sets the learning rate to be one over the number of iterations. Finally, [39] uses off-trajectory training. To remove the impact of the training trajectories, our experiments include both off-trajectory and on-trajectory training.

Results. Tables 4.1 and 4.2 show the results on the ACE04 and MUC-6 corpora repsectively. For metrics, we compute MUC [40], B^3 [1], CEAF [25], and CoNLL F1 [30], all of which have been employed in the CoNLL Shared-task 2011. For MUC, B^3, and CEAF, we show the precision, recall and F1 measure separately. Note that CoNLL F1 is simply the average F1 values of the other three metrics. From the tables, we can see that our approach outperforms BGBB for the ACE corpus for all three metrics with both on-trajectory and off-trajectory training. For the MUC6 corpus, our method outperforms BGBB for off-trajectory training and performs similarly for the on-trajectory setting.

4.3 Joint entity and event coreference resolution across documents

Cross-document joint entity and event coreference resolution is a challenging problem that involves resolving the coreferences for entities (noun phrases) and events (verbs) across multiple

	MUC			B^3			CEAF-ϕ_4			BLANC			
Set.	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1
Enti.	72.78	83.14	77.62	56.62	81.22	66.72	53.45	37.42	44.02	69.84	83.78	74.98	62.79
Even.	65.95	70.13	67.98	49.19	75.75	59.65	39.09	34.70	36.76	67.11	72.29	69.35	54.80
Both	70.81	86.04	77.69	54.46	83.74	65.99	60.57	35.79	44.99	69.24	85.59	74.96	62.89

Table 4.4: Cross-document joint entity and event coreference resolution (on-trajectory training) on the EECB corpus using BGBB approach.

	MUC			B^3			CEAF-ϕ_4			BLANC			
Set.	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1
Enti.	74.83	84.45	79.35	58.45	82.07	68.27	55.26	39.47	46.05	70.93	84.94	76.16	64.56
Even.	65.55	71.08	68.20	50.13	77.99	61.03	39.28	33.60	36.22	68.46	74.68	71.12	55.15
Both	72.15	87.30	79.01	56.03	85.08	67.56	62.23	36.81	46.26	70.39	87.08	76.30	64.28

Table 4.5: Cross-document joint entity and event coreference resolution (off-trajectory training) on the EECB corpus using BGVB approach.

documents simultaneously.

Data. We employ the benchmark EECB corpus [24] for our experiments. As an extension to the ECB corpus created by [2], EECB contains 482 documents, which are clustered into 43 topics. We use the same split for training, validation, and testing as [24]. That is, out of 43 topics, we use 12 topics for training, 3 topics for validation, and 28 topics for testing. The detailed information about the EECB corpus is shown below:

Features. We employ the same set of features as [24] with two minor distinctions. First, in addition to the merge actions, we introduce the HALT action to serve the role of a terminal state for Easy-first search. Following the convention of [39], we represent the HALT action by a feature vector of all zeros except for a halt feature that is set to 1. For all other actions, we set the halt feature to zero. Note that the learned weight of this halt feature operates as a threshold on action scores, for which [24] manually specified a fixed value of 0.5. In the inference stage, if no merge action scores higher than this threshold, the search procedure terminates. Another minor distinction is that we employed different semantic role labeling (SRL) software due to an availability issue. The SRL software is trained on both NomBank and PropBank [22].

Baselines. We compare our approach against two baseline methods. The first is the current state-of-the-art cross-document joint entity and event coreference system by [24]. The second is the Best Good Best Bad (BGBB) update strategy, which can be viewed as an application of the Easy-first coreference system [39] to the joint coreference problem.

	MUC			B^3			CEAF-ϕ_4			BLANC			
Set.	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1
Enti.	71.18	83.99	77.06	53.14	83.79	65.03	56.19	36.49	44.24	67.32	87.50	73.67	62.11
Even.	64.44	77.42	70.34	45.75	82.42	58.84	47.90	33.42	39.37	64.89	87.22	71.21	56.18
Both	69.24	86.71	76.99	50.99	86.15	64.06	62.11	34.58	44.43	66.77	88.96	73.42	61.83

Table 4.6: Cross-document joint entity and event coreference resolution (off-trajectory training) on the EECB corpus using BGBB approach.

	MUC			B^3			CEAF-ϕ_4			BLANC			
Set.	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1
Enti.	68.07	90.07	77.54	52.11	90.20	66.05	60.81	30.43	40.56	67.76	90.16	74.59	61.38
Even.	65.15	82.37	72.75	50.27	86.92	63.70	53.43	32.76	40.62	69.15	91.68	76.18	59.02
Both	67.23	91.72	77.59	51.57	91.51	65.97	66.37	30.48	41.77	68.10	91.94	75.26	61.78

Table 4.7: Cross-document joint entity and event coreference resolution (off-trajectory training) on the EECB corpus using Stanford system.

Experimental setup. We implement our approach and the BGBB baseline on top of the Stanford multi-pass sieve system [31]. All three methods employ the same set of features and apply the same initial processing step to the noun-phrase mentions as described in [24]. We set up our experiments to closely resemble the experiments in [24]. The parameters of BGVB (λ, CCCP iterations T, and gradient descent iterations t) and BGBB (learning rate η and maximum updates per ieration k) are tuned with five-fold cross-validation within the training set. For both BGVB and BGBB, we tune the halt feature using the validation set to determine the stopping condition for inference. For the method of [24], we use the implementation provided by the author and follow the parameter setup suggested by the original paper.

Results. Our experiments consider both on-trajectory and off-trajectory training. [24] performs offline training, where the training examples can be viewed as collected in an off-trajectory fashion (not restricted to taking good actions during training). Thus it is omitted from comparison for the on-trajectory setting. We present on-trajectory performance in Table 4.3 and 4.4, and off-trajectory performance in Table 4.5, 4.6 and 4.7. We measure the performances using the same set of metrics as for within-document entity coreference (MUC, B^3, CEAF, CoNLL F1) along with an additional metric BLANC [34], which has been also reported by [24].

We present the performance measures in three different settings: evaluating only entity clusters, evaluating only event clusters, and evaluating both types of clusters. From Table 4.3 and 4.4, we observe that with on-trajectory training our approach consistently outperforms BGBB for all

	Train	**Dev**	**Test**	**Total**
# Topics	12	3	28	43
# Documents	112	39	331	482
# Entities	459	46	563	1068
# Entity Mentions	1723	259	3465	5447
# Events	300	30	444	774
# Event Mentions	751	140	1642	2553

Table 4.8: Corpus Statistics.

three settings under all performance measures. The results of our approach are almost 2 percent CoNLL F1 higher than corresponding results of BGBB. It indicates that our approach behaves better than BGBB by capturing the learning goal of Easy-first framework. For off-trajectory training (Table 4.5, 4.6 and 4.7), our approach performs decisively better than BGBB as well as the method by [24] on entity clusters. Compared with the joint co-reference system, the results of our approach are also significantly better in the Entity and Both setting. Our approach does not perform well on Event setting. The verdict on the event clusters is less clear, with different measures favoring different methods. When jointly considering both entities and events, our method outperforms the other two methods consistently across the measures due to the larger number of entity mentions. It can be noted that different system focus on different aspect. One system does better on entity setting, may not do better on event setting. So it is very important to balance the trade-off between both aspects.

4.4 Summary of results

Our evaluation demonstrates that our BGVB approach consistently outperforms the baseline methods for both problem domains and under two different training regimes (on-trajectory and off-trajectory). The performance difference is more pronounced for the cross-document joint coreference problem for which our method outperforms the current state-of-the-art by a consistent 2-3 percentage points (for multiple performance measures). We did not observe any significant or consistent difference between on-trajectory versus off-trajectory training. In the following section, we take a closer look at the behavior of BGVB compared to BGBB to gain a better understanding of why it performs better.

4.5 Discussion

There are two key distinctions between BGVB and the BGBB update rule. First, BGBB considers only the best-scoring bad action in its update, whereas our method considers all bad actions that are causing constraint violations (in Equation 3.2) in each update. Second, our method follows a passive-aggressive strategy to discourage overly-aggressive updates. Our hypothesis is that these distinctions allow us to introduce more stability in learning and help avoid overfitting to specific bad actions encountered during training.

Corpus	Approach	Total Steps	Mistakes	Recoveries	Percentage
ACE04	BGVB	50195	16228	4255	0.262
	BGBB	50195	11625	4075	0.351
MUC-6	BGVB	40975	16436	3579	0.218
	BGBB	40975	14727	4156	0.282

Table 4.9: Training statistics on MUC-6 and ACE04 corpora

	Approach	Mean	Variance	STDEV
ACE04	BGVB	0.87	0.0047	0.069
	BGBB	0.82	0.0064	0.080
MUC-6	BGVB	0.86	0.0055	0.074
	BGBB	0.80	0.0152	0.123

Table 4.10: Global performance of the learned weights.

To shed some light on this hypothesis, we take a closer look at the effect of the updates during training. Focusing on within-document coreference and on-trajectory training, we collected some interesting statistics during five iterations of training for both methods, which are presented in Table 4.9. The third column of the table records the total number of search steps. The fourth column shows the number of mistakes (bad actions chosen during search) encountered in the process of training, each incurring a round of updates. The next two columns show the number of times that the update is successful ("recoveries", where the highest-scoring action is good after update) and its corresponding percentage.

We were surprised to note that our method (BGVB) is significantly less successful at correcting mistakes. How could it be less effective in correcting the mistakes but more effective overall? The explanation lies in the passive-aggressive element of our objective, which explicitly encourages small changes to the weights, sometimes at the expense of not satisfying all the constraints. This explains why our method tends to fail more at correcting the mistakes, but does

not provide an answer to why its overall performance tends to be better.

To answer this question, we further examine the quality of the weights obtained by both methods in a global setting. That is, we recorded all the weights that are produced by the two learning algorithms and evaluated how well each weight can guide the greedy search on the training set. To do this, we randomly generated some sample gold search trajectories (by randomly choosing good action at each search step) on the training data, and at each search step evaluated the learned weights to choose actions. If a weight chose a good action, it was considered a correct decision. For each method, we computed the percentage of correct decisions made by all of the learned weights, averaged across five randomly-generated search trajectories. The mean and variance are reported in Table 4.10. The results show that the weights learned by BGVB have consistently better global performance and smaller variance. This suggests that by satisfying more local constraints, BGBB is indeed suffering from overfitting, to which our method is less prone.

Chapter 5: Conclusions and Future Work

Learning problem is a challenge within the Easy-first framework due to its difficulty to formulate. The existed methods are either rule-based or in a heuristic manner. We proposed a novel online learning algorithm for the Easy-first framework. By formulating the learning problem as an optimization objective, we capture the essence of the learning goal for the Easy-first framework: select the best scoring action at each search state while avoiding overly-aggressive updates.

Experiments on two NLP domains, within-document coreference resolution and cross-document joint entity and event coreference resolution, showed that our greedy learning method outperforms an existing Easy-first training method and the current state-of-the-art for both problem domains in terms of all evaluation metrics, including MUC, B-Cubed, CEAF-ϕ_4, BLANC and CONLL F-score. As we know, different evaluation metrics evaluate different aspects of the coreference resolution tasks. Hence, it tends to show that our proposed approach learns a better greedy policy than other approaches in the Easy-first framework.

We provide several interesting ideas to extend our work. As we know, Easy-first framework works in a greedy search space. Sometimes greedy decisions are hard to make, so we can consider searching in limited-discrepancy search (LDS) space [12] via HC-Search approach [13]. Sparse versions of LDS space [14] can be employed to improve the efficiency of the search process. The second idea is to apply our method on the off-trajectory training regime. As we know, we just can encounter those states appearing in the path from the initial state to the gold state in the training phase. So the learned policy will have no clue to behave in front of an unseen state during the testing phase. This way, it will lead our search into a huge different trajectory which may degrade the performance of our approach. Unlike our current implementation of off-trajectory, it may be more reasonable to use a more advanced mechanism to do this. For every search step, we consider pairwise actions from each state. This way, there exist a large number of candidates to consider for each step. Especially, the learning problem for a lot of NLP applications is highly inseparable. Hence, the third idea is to take advantage of prune method to prune a lot of actions while to maintain our method's accuracy.

Bibliography

[1] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566, 1998.

[2] Cosmin Adrian Bejan and Sanda Harabagiu. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1412–1422, 2010.

[3] Eric Bengtson and Dan Roth. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 294–303, 2008.

[4] Chen Chen and Vincent Ng. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, CoNLL '12, pages 56–63, 2012.

[5] Zheng Chen and Heng Ji. Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-4, pages 54–57, 2009.

[6] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, December 2006.

[7] Aron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. First-order probabilistic models for coreference resolution. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, pages 81–88, 2007.

[8] Hal Daumé III, John Langford, and Daniel Marcu. Search-based structured prediction. *MLJ*, 75(3):297–325, 2009.

[9] Hal Daumé III and Daniel Marcu. Learning as search optimization: Approximate large margin methods for structured prediction. In *ICML*, 2005.

[10] Pascal Denis and Jason Baldridge. Specialized models and ranking for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 660–669, 2008.

[11] Thomas G. Dietterich, Hermann Hild, and Ghulum Bakiri. A comparison of ID3 and backpropagation for english text-to-speech mapping. *MLJ*, 18(1):51–80, 1995.

[12] J. R. Doppa, A. Fern, and P. Tadepalli. Output space search for structured prediction. In *ICML*, 2012.

[13] J. R. Doppa, A. Fern, and P. Tadepalli. HC-Search: Learning heuristics and cost functions for structured prediction. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI-2013)*, 2013.

[14] J. R. Doppa, A. Fern, and P. Tadepalli. Structured prediction via output space search. *Journal of Machine Learning Research*, 15, 2014.

[15] Yoav Goldberg and Michael Elhadad. An efficient algorithm for easy-first non-directional dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 742–750, 2010.

[16] Barbara J. Grosz. The representation and use of focus in a system for understanding dialogs. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, pages 67–76, 1977.

[17] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, ACL '83, pages 44–50, 1983.

[18] Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Comput. Linguist.*, 21(2):203–225, June 1995.

[19] Aria Haghighi and Dan Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 385–393, 2010.

[20] Tian He. Coreference resolution on entities and events for hospital discharge summaries. Technical report, 2007.

[21] Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. Event coreference for information extraction. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, ANARESOLUTION '97, pages 75–81, 1997.

[22] Richard Johansson and Pierre Nugues. Dependency-based syntactic-semantic analysis with propbank and nombank. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, CoNLL '08, pages 183–187, 2008.

[23] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 28–34, 2011.

[24] Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 489–500, 2012.

[25] Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 25–32, 2005.

[26] Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, 2004.

[27] Francis Maes, Ludovic Denoyer, and Patrick Gallinari. Structured prediction with reinforcement learning. *Machine Learning*, 77(2-3):271–301, 2009.

[28] Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1396–1411, 2010.

[29] Hoifung Poon and Pedro Domingos. Joint unsupervised coreference resolution with markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 650–659, 2008.

[30] Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 1–27, 2011.

[31] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 492–501, 2010.

[32] Altaf Rahman and Vincent Ng. Narrowing the modeling gap: A cluster-ranking approach to coreference resolution. *J. Artif. Int. Res.*, 40(1):469–521, January 2011.

[33] William M. Rand. Objective criteria for the evaluation of clustering methods. 66(336):846–850, 1971.

[34] M. Recasens and E. Hovy. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17:485–510, 10 2011.

[35] Stéphane Ross, Geoffery Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pages 627–635, 2011.

[36] L. Shen, G. Satta, and A. Joshi. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.

[37] Candace L Sidner. Towards a computational theory of definite anaphora comprehension in english discourse. Technical report, 1979.

[38] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544, December 2001.

[39] Veselin Stoyanov and Jason Eisner. Easy-first coreference resolution. In *COLING*, pages 2519–2534, 2012.

[40] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, pages 45–52, 1995.

[41] Yuehua Xu, Alan Fern, and Sung Wook Yoon. Learning linear ranking functions for beam search with application to planning. *Journal of Machine Learning Research*, 10:1571–1610, 2009.

[42] A. L. Yuille and Anand Rangarajan. The concave-convex procedure. *Neural Comput.*, 15(4):915–936, April 2003.