Training Conditional Random Fields Using Incomplete Annotations

Yuta Tsuboi, Hisashi Kashima
Tokyo Research Laboratory,
IBM Research, IBM Japan, Ltd
Yamato, Kanagawa 242-8502, Japan
{yutat,hkashima}@jp.ibm.com

Shinsuke Mori
Academic Center for Computing and
Media Studies, Kyoto University
Sakyo-ku, Kyoto 606-8501, Japan
forest@i.kyoto-u.ac.jp

Hiroki Oda
Shinagawa, Tokyo, Japan
oda@fw.ipsj.or.jp

Yuji Matsumoto
Graduate School of Information Science,
Nara Institute of Science and Technology
Takayama, Ikoma, Nara 630-0101, Japan
matsu@is.naist.jp

Abstract

We address corpus building situations, where complete annotations to the whole corpus is time consuming and unrealistic. Thus, annotation is done only on crucial part of sentences, or contains unresolved label ambiguities. We propose a parameter estimation method for Conditional Random Fields (CRFs), which enables us to use such incomplete annotations. We show promising results of our method as applied to two types of NLP tasks: a domain adaptation task of a Japanese word segmentation using partial annotations, and a part-of-speech tagging task using ambiguous tags in the Penn treebank corpus.

1 Introduction

Annotated linguistic corpora are essential for building statistical NLP systems. Most of the corpora that are well-known in NLP communities are completely-annotated in general. However it is quite common that the available annotations are partial or ambiguous in practical applications. For example, in domain adaptation situations, it is time-consuming to annotate all of the elements in a sentence. Rather, it is efficient to annotate certain parts of sentences which include domain-specific expressions. In Section 2.1, as an example of such efficient annotation, we will describe the effectiveness of partial annotations in the domain adaptation task for Japanese word segmentation (JWS). In addition, if the annotators are domain experts rather than linguists, they are unlikely to be confident about the annotation policies and may prefer to be allowed to defer some linguistically complex decisions. For many NLP tasks, it is sometimes difficult to decide which label is appropriate in a particular context. In Section 2.2, we show that such ambiguous annotations exist even in a widely used corpus, the Penn treebank (PTB) corpus.

This motivated us to seek to incorporate such incomplete annotations into a state of the art machine learning technique. One of the recent advances in statistical NLP is Conditional Random Fields (CRFs) (Lafferty et al., 2001) that evaluate the global consistency of the complete structures for both parameter estimation and structure inference, instead of optimizing the local configurations independently. This feature is suited to many NLP tasks that include correlations between elements in the output structure, such as the interrelation of part-of-speech (POS) tags in a sentence. However, conventional CRF algorithms require fully annotated sentences. To incorporate incomplete annotations into CRFs, we extend the structured output problem in Section 3. We focus on partial annotations or ambiguous annotations in this paper. We also propose a parameter estimation method for CRFs using incompletely annotated corpora in Section 4. The proposed method marginalizes out the unknown labels so as to optimize the likelihood of a set of possible label structures which are consistent with given incomplete annotations.

We conducted two types of experiments and observed promising results in both of them. One was a domain adaptation task for JWS to assess the proposed method for partially annotated data. The other was a POS tagging task using ambiguous annotations that are contained in the PTB corpus. We summarize related work in Section 6, and conclude

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (http://creativecommons.org/licenses/by-nc-sa/3.0/). Some rights reserved.

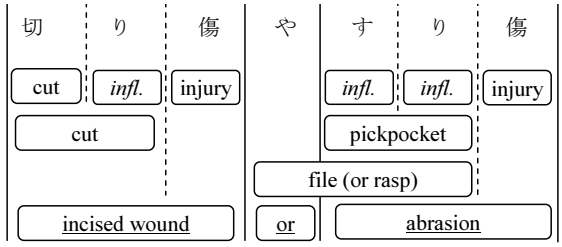

Figure 1: An example of word boundary ambiguities: *infl.* stands for an inflectional suffix of a verb.

in Section 7.

2 Incomplete Annotations

2.1 Partial Annotations

In this section, we describe an example of an efficient annotation which assigns partial word boundaries for the JWS task.

It is not trivial to detect word boundaries for non-segmented languages such as Japanese or Chinese. For example, the correct segmentation of the Japanese phrase "切り傷やすり傷" (incised wound or abrasion) is shown by the lowest boxes segmented by the solid lines in Figure 1. However, there are several overlapping segmentation candidates, which are shown by the other boxes, and possible segmentation by the dashed lines. Thus, the decisions on the word segmentation require considering the context, so simple dictionary lookup approach is not appropriate. Therefore statistical methods have been successfully used for JWS tasks. Previous work (Kudo et al., 2004) showed CRFs outperform generative Markov models and discriminative history-based methods in JWS. In practice, a statistical word segment analyzer tends to perform worse for text from different domains, so that additional annotations for each target domain are required. A major cause of errors is the occurrence of unknown words. For example, if "すり傷" (abrasion) is an unknown word, the system may accept the word sequence of "切り傷やすり傷" as "切り傷" (incised wound), "やすり" (file), and "傷" (injury) by mistake.

On one hand, lists of new terms in the target domain are often available in the forms of technical term dictionaries, product name lists, or other sources. To utilize those domain word lists, Mori (2006) proposed a KWIC (KeyWord In Context) style annotation user interface (UI) with which a user can delimit a word in a context with a single user action. In Figure 2, an annotator marks the occurrences of "すり傷", a word in the domain word

	感染、角膜のこ	すり傷	、角膜潰瘍、
○	皮膚に切り傷や	すり傷	を負った場合
○	泥まみれの深い	すり傷	や、皮下深く

Figure 2: An example of KWIC style annotation: marked lines are identified as a correct segmentation.

list, if they are used as a real word in their context. The "すり傷" in the first row is a part of another word "こすり傷" (scratch), and the annotator marks the last two rows as correctly segmented examples. This UI simplifies annotation operations for segmentation to yes/no decisions, and this simplification can also be effective for the reduction of the annotation effort for other NLP tasks. For example, the annotation operations for unlabeled dependency parsing can be simplified into a series of yes/no decisions as to whether or not given two words have syntactic dependency. Compared with sentence-wise annotation, the partial annotation is not only effective in terms of control operations, but also reduces annotation errors because it does not require annotating the word boundaries that an annotator is unsure of. This feature is crucial for annotations made by domain experts who are not linguists. [1] We believe partial annotation is effective in creating corpora for many other structured annotations in the context of the domain adaptations.

2.2 Ambiguous Annotations

Ambiguous annotations in this paper refer to a set of candidate labels annotated for a part of a structured instance. For example, the following sentence from the PTB corpus includes an ambiguous annotation for the POS tag of "pending":

That/DT suit/NN is/VBZ pending/VBG|JJ ./. ,

where words are paired with their part-of-speech tag by a forward slash ("/").[2] Uncertainty concerning the proper POS tag of "pending" is represented by the disjunctive POS tag ("VBG and JJ") as indicated by a vertical bar.

The existence of the ambiguous annotations is due to the task definition itself, the procedure man-

[1] The boundary policies of some words are different even among linguists. In addition, the boundary agreement is even lower in Chinese (Luo, 2003).

[2] These POS tags used here are DT:determiner, NN:common noun, VBZ:present tense 3rd person singular verb, VBG:gerund or present participle verb, JJ:adjective, NNS:plural noun, RBR:comparative adverb, IN:preposition or subordinating conjunction, and RB:adverb.

frequency	word	POS tags
15	data	NN|NNS
10	more	JJR|RBR
7	pending	JJ|VBG
4	than	IN|RB

Table 1: Words in the PTB with ambiguous POSs.

ual for the annotators, or the inadequate knowledge of the annotators. Ideally, the annotations should be disambiguated by a skilled annotator for the training data. However, even the PTB corpus, whose annotation procedure is relatively well-defined, includes more than 100 sentences containing POS ambiguities such as those listed in Table 1. Although the number of ambiguous annotations is not considerably large in PTB corpus, corpora could include more ambiguous annotations when we try to build wider coverage corpora. Also, ambiguous annotations are more common in the tasks that deal with semantics, such as information extraction tasks so that learning algorithms must deal with ambiguous annotations.

3 Problem Definition

In this section, we give a formal definition of the supervised structured output problem that uses partial annotations or ambiguous annotations in the training phase. Note that we assume the input and output structures are sequences for the purpose of explanation, though the following discussion is applicable to other structures, such as trees.

Let $\boldsymbol{x}=(x_1, x_2, \cdots, x_T)$ be a sequence of observed variables $x_t \in X$ and $\boldsymbol{y}=(y_1, y_2, \cdots, y_T)$ be a sequence of label variables $y_t \in Y$. Then the supervised structured output problem can be defined as learning a map $\boldsymbol{X} \rightarrow \boldsymbol{Y}$. In the Japanese word segmentation task, $\boldsymbol{x}$ can represent a given sequence of character boundaries and $\boldsymbol{y}$ is a sequence of the corresponding labels, which specify whether the current position is a word boundary.[3] In the POS tagging task, $\boldsymbol{x}$ represents a word sequence and $\boldsymbol{y}$ is a corresponding POS tag sequence. An incomplete annotation, then, is defined as a sequence of subset of the label set instead of a sequence of labels. Let $\boldsymbol{L}=(L_1, L_2, \cdots, L_T)$ be a sequence of label subsets for an observed sequence $\boldsymbol{x}$, where $L_t \in 2^Y - \{\emptyset\}$. The partial annotation at position s is where L_s is a singleton and the rest $L_{t \neq s}$ is $\boldsymbol{Y}$. For example, if a sentence with 6 character boundaries (7 characters) is partially annotated using the KWIC UI described in Section 2.1, a word annotation where its boundary begins with $t = 2$ and ends with $t = 5$ will be represented as:

$$\boldsymbol{L} = (\{\bigcirc, \times\}, \underbrace{\{\bigcirc\}, \{\times\}, \{\times\}, \{\bigcirc\}}_{\text{partial annotation}}, \{\bigcirc, \times\}),$$

where $\bigcirc$ and $\times$ denote the word boundary label and the non-word boundary label, respectively. The ambiguous annotation is represented as a set which contains candidate labels. The example sentence including the ambiguous POS tag in Section 2.2 can be represented as:

$$\boldsymbol{L} = (\{\text{DT}\}, \{\text{NN}\}, \{\text{VBZ}\}, \underbrace{\{\text{VBG}, \text{JJ}\}}_{\text{ambiguous annotation}}, \{.\}).$$

Note that, if all the elements of a given sequence are annotated, it is the special case such that the size of all elements is one, i.e. $|L_t| = 1$ for all $t = 1, \cdots, T$. The goal in this paper is training a statistical model from partially or ambiguously annotated data, $D = \{(\boldsymbol{x}^{(n)}, \boldsymbol{L}^{(n)})\}_{n=1}^{N}$.

4 Marginalized Likelihood for CRFs

In this section, we propose a parameter estimation procedure for the CRFs (Lafferty et al., 2001) incorporating partial or ambiguous annotations. Let $\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{X} \times \boldsymbol{Y} \rightarrow \Re^d$ denote a map from a pair of an observed sequence $\boldsymbol{x}$ and a label sequence $\boldsymbol{y}$ to an arbitrary feature vector of d dimensions, and $\boldsymbol{\theta} \in \Re^d$ denotes the vector of the model parameters. CRFs model the conditional probability of a label sequence $\boldsymbol{y}$ given an observed sequence $\boldsymbol{x}$ as:

$$P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{e^{\boldsymbol{\theta} \cdot \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y})}}{Z_{\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{Y}}}, \tag{1}$$

where $\cdot$ denotes the inner product of the vectors, and the denominator is the normalization term that guarantees the model to be a probability:

$$Z_{\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{S}} = \sum_{\boldsymbol{y} \in \boldsymbol{S}} e^{\boldsymbol{\theta} \cdot \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y})}.$$

Then once $\boldsymbol{\theta}$ has been estimated, the label sequence can be predicted by $\hat{\boldsymbol{y}} = \text{argmax}_{\boldsymbol{y} \in \boldsymbol{Y}} P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$. Since the original CRF learning algorithm requires a completely labeled sequence $\boldsymbol{y}$, the incompletely annotated data $(\boldsymbol{x}, \boldsymbol{L})$ is not directly applicable to it.

[3] Peng et al. (2004) defined the word segmentation problem as labeling each character as whether or not the previous character boundary of the current character is a word boundary. However, we employ our problem formulation since it is redundant to assign the first character of a sentence as the word boundary in their formulation.

Let $\boldsymbol{Y_L}$ denote all of the possible label sequence consistent with $\boldsymbol{L}$. We propose to use the conditional probability of the subset $\boldsymbol{Y_L}$ given $\boldsymbol{x}$:

$$P_{\boldsymbol{\theta}}(\boldsymbol{Y_L}|\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \boldsymbol{Y_L}} P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}), \quad (2)$$

which marginalizes the unknown $\boldsymbol{y}$s out. Then the maximum likelihood estimator for this model can be obtained by maximizing the log likelihood function:

$$\mathrm{LL}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln P_{\boldsymbol{\theta}}(\boldsymbol{Y}_{\boldsymbol{L}^{(n)}}|\boldsymbol{x}^{(n)}) \quad (3)$$
$$= \sum_{n=1}^{N} \left(\ln Z_{\boldsymbol{\theta},\boldsymbol{x}^{(n)},\boldsymbol{Y}_{\boldsymbol{L}^{(n)}}} - \ln Z_{\boldsymbol{\theta},\boldsymbol{x}^{(n)},\boldsymbol{Y}} \right).$$

This modeling naturally embraces label ambiguities in the incomplete annotation.[4]

Unfortunately, equation (3) is not a concave function[5] so that there are local maxima in the objective function. Although this non-concavity prevents efficient global maximization of equation (3), it still allows us to incorporate incomplete annotations using gradient ascent iterations (Sha and Pereira, 2003). Gradient ascent methods require the partial derivative of equation (3):

$$\frac{\partial \mathrm{LL}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{n=1}^{N} \left(\sum_{\boldsymbol{y} \in \boldsymbol{Y}_{\boldsymbol{L}^{(n)}}} P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{Y}_{\boldsymbol{L}^{(n)}}, \boldsymbol{x}^{(n)}) \boldsymbol{\Phi}(\boldsymbol{x}^{(n)}, \boldsymbol{y}) \right.$$
$$\left. - \sum_{\boldsymbol{y} \in \boldsymbol{Y}} P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}^{(n)}) \boldsymbol{\Phi}(\boldsymbol{x}^{(n)}, \boldsymbol{y}) \right), \quad (4)$$

where

$$P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{Y_L}, \boldsymbol{x}) = \frac{e^{\boldsymbol{\theta} \cdot \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y})}}{Z_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{Y_L}}} \quad (5)$$

is a conditional probability that is normalized over $\boldsymbol{Y_L}$.

Equations (3) and (4) include the summations of all of the label sequences in $\boldsymbol{Y}$ or $\boldsymbol{Y_L}$. It is not practical to enumerate and evaluate all of the label configurations explicitly, since the number of all of the possible label sequences is exponential on the number of positions t with $|L_t| > 1$. However, under the Markov assumption, a modification of the *Forward-Backward algorithm* guarantees polynomial time computation for the equations (3) and (4). We explain this algorithm in Appendix A.

[4] It is common to introduce a prior distribution over the parameters to avoid over-fitting in CRF learning. In the experiments in Section 5, we used a Gaussian prior with the mean 0 and the variance σ^2 so that $-\frac{||\boldsymbol{\theta}||^2}{2\sigma^2}$ is added to equation (3).

[5] Since its second order derivative can be positive.

	domain	#sentences	#words
(A)	conversation	11,700	145,925
(B)	conversation	1,300	16,348
(C)	medical manual	1,000	29,216

Table 2: Data statistics.

Types	Template
Characters	$c_{-1}, c_{+1},$
Character types	$c_{-2}c_{-1}, c_{-1}c_{+1}, c_{+1}c_{+2},$
Term in dic.	$c_{-2}c_{-1}c_{+1}, c_{-1}c_{+1}c_{+2}$
Term in dic. starts at	c_{-1}, c_{+1}
Term in dic. ends at	

Table 3: Feature templates: Each subscript stands for the relative distance from a character boundary.

5 Experiments

We conducted two types of experiments, assessing the proposed method in 1) a Japanese word segmentation task using partial annotations and 2) a POS tagging task using ambiguous annotations.

5.1 Japanese Word Segmentation Task

In this section, we show the results of domain adaptation experiments for the JWS task to assess the proposed method. We assume that only partial annotations are available for the target domain. In this experiment, the corpus for the source domain is composed of example sentences in a dictionary of daily conversation (Keene et al., 1992). The text data for the target domain is composed of sentences in a medical reference manual (Beers, 2004) . The sentences of all of the source domain corpora (A), (B) and a part of the target domain text (C) were manually segmented into words (see Table 2).

The performance measure in the experiments is the standard F measure score, $F = 2RP/(R+P)$ where

$$R = \frac{\text{\# of correct words}}{\text{\# of words in test data}} \times 100$$
$$P = \frac{\text{\# of correct words}}{\text{\# of words in system output}} \times 100.$$

In this experiment, the performance was evaluated using 2-fold cross-validation that averages the results over two partitions of the data (C) into the

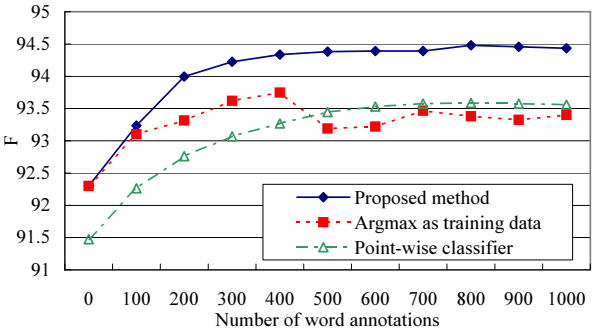

Figure 3: Average performances varying the number of word annotations over 2 trials.

data for annotation and training (C1) versus the data for testing (C2).

We implemented first order Markov CRFs. As the features for the observed variables, we use the characters and character type n-gram (n=1, 2, 3) around the current character boundary. The character types are categorized into *Hiragana*, *Katakana*, *Kanji*, English alphabet, Arabic numerals, and symbols. We also used lexical features consulting a dictionary: one is to check if any of the above defined character n-grams appear in a dictionary (Peng et al., 2004), and the other is to check if there are any words in the dictionary that start or end at the current character boundary. We used the *unidic*[6] (281K distinct words) as the general purpose dictionary, and the *Japanese Standard Disease Code Master* (JSDCM)[7] (23K distinct words) as the medical domain dictionary. The templates for the features we used are summarized in Table 3. To reduce the number of parameters, we selected only frequent features in the source domain data (A) or in about $50K$ of the unsegmented sentences of the target domain.[8] The total number of distinct features was about $300K$.

A CRF that was trained using only the source domain corpus (A), CRF$^{\mathrm{S}}$, achieved F=96.84 in the source domain validation data (B). However, it showed the need for the domain adaptation that this CRF^{S} suffered severe performance degradation (F=92.3) on the target domain data. This experiment was designed for the case in which a user selects the occurrences of words in the word list using the KWIC interface described in Section 2.1. We employed JSDCM as a word list in which 224 distinct terms appeared on average over 2 test sets (C1). The number of word annotations varied from 100 to 1000 in this experiment. We prioritized the occurrences of each word in the list using a selective sampling technique. We used label entropy (Anderson et al., 2006), $H(\boldsymbol{y}_t^s) = \sum_{\boldsymbol{y}_t^s \in \boldsymbol{Y}_t^s} P_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{y}_t^s|\boldsymbol{x}) \ln P_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{y}_t^s|\boldsymbol{x})$, as importance metric of each word occurrence, where $\tilde{\boldsymbol{\theta}}$ is the model parameter of CRF$^{\mathrm{S}}$, and $\boldsymbol{y}_t^s = (y_t, y_{t+1}, \cdots, y_s) \in \boldsymbol{Y}_t^s$ is a subsequence starting at t and ending at s in $\boldsymbol{y}$. Intuitively, this metric represents the prediction confidence of CRF$^{\mathrm{S}}$.[9] As training data, we mixed the complete annotations (A) and these partial annotations on data (C1) because that performance was better than using only the partial annotations.

We used *conjugate gradient* method to find the local maximum value with the initial value being set to be the parameter vector of CRF^{S}. Since the amount of annotated data for the target domain was limited, the hyper-parameter σ was selected using the corpus (B).

For the comparison with the proposed method, the CRFs were trained using the most probable label sequences consistent with $\boldsymbol{L}$ (denoted as *argmax*). The most probable label sequences were predicted by the CRF$^{\mathrm{S}}$. Also, we used a *point-wise classifier*, which independently learns/classifies each character boundary and just ignores the unannotated positions in the learning phase. As the point-wise classifier, we implemented a maximum entropy classifier which uses the same features and optimizer as CRFs.

Figure 3 shows the performance comparisons varying the number of word annotations. The combination of both the proposed method and the selective sampling method showed that a small number of word annotations effectively improved the word segmentation performance. In addition, the proposed method significantly outperformed *argmax* and *point-wise classifier* based on the *Wilcoxon signed rank test* at the significance level of 5%. This result suggests that the proposed method maintains CRFs' advantage over the *point-wise classifier* and properly incorporates partial annotations.

5.2 Part-of-speech Tagging Task

In this section, we show the results of the POS tagging experiments to assess the proposed method using ambiguous annotations.

[6] Ver. 1.3.5; http://www.tokuteicorpus.jp/dist/

[7] Ver. 2.63; http://www2.medis.or.jp/stdcd/byomei/

[8] The data (B) and (C), which were used for validation and test, were excluded from this feature selection process.

[9] We selected word occurrences in a batch mode since each training of the CRFs takes too much time for interactive use.

	Ex.1	Ex.2
ambiguous sentences (training)	118	
unique sentences (training)	1,480	2,960
unique sentences (test)	11,840	

Table 4: Training and test data for POS tagging.

As mentioned in Section 2.2, there are words which have two or more candidate POS tags in the PTB corpus (Marcus et al., 1993). In this experiment, we used 118 sentences in which some words (82 distinct words) are annotated with ambiguous POS tags, and these sentences are called the *POS ambiguous sentences*. On the other hand, we call sentences in which the POS tags of these terms are uniquely annotated as the *POS unique sentences*.

The goal of this experiment is to effectively improve the tagging performance using both these POS ambiguous sentences and the POS unique sentences as the training data. We assume that the amount of training data is not sufficient to ignore the POS ambiguous sentences, or that the POS ambiguous sentences make up a substantial portion of the total training data. Therefore we used a small part ($1/10$ or $1/5$) of the POS unique sentences for training the CRFs and evaluated their performance using other ($4/5$) POS unique sentences. We conducted two experiments in which different numbers of unique sentences were used in the training phases, and these settings are summarized in Table 4.

The feature sets for each word are the case-insensitive spelling, the orthographic features of the current word, and the sentence's last word. The orthographic features are whether a spelling begins with a number or an upper case letter; whether it begins with an upper case letter and contains a period (“.”); whether it is all upper case letters or all lower case letters; whether it contains a punctuation mark or a hyphen; and the last one, two, and three letters of the word. Also, the sentence's last word corresponds to a punctuation mark (e.g. “.”, “?”, “!”). We employed only features that appeared more than once. The total number of resulting distinct features was about $14K$. Although some symbols are treated as distinct tags in the PTB tag definitions, we aggregated these symbols into a symbol tag (SYM) since it is easy to restore original symbol tags from the SYM tag. Then, the number of the resulting tags was 36.

For the comparison with the proposed method (*mrg*), we used three heuristic rules that disambiguated the annotated candidate POS tags in the POS ambiguous sentences. These rules selected a POS tag 1) at *random*, 2) as the *first* one in the description order[10], 3) as the most *frequent* tag in the corpus. In addition, we evaluated the case when the POS ambiguous sentences are 4) *discarded* from the training data.

For evaluation, we employed the Precision (P) and Average Precision for Ambiguous words (APA):

$$\mathrm{P} = \frac{\text{\# of correctly tagged word}}{\text{\# of all word occurrences}} \times 100,$$

$$\mathrm{APA} = \frac{1}{|\boldsymbol{A}|} \sum_{w \in \boldsymbol{A}} \frac{\text{\# of the correctly tagged } w}{\text{\# of all occurrences of } w} \times 100,$$

where $\boldsymbol{A}$ is a word set and is composed of the word for which at least one of its occurrences is ambiguously annotated. Here, we employed APA to evaluate each ambiguous words equally, and $|\boldsymbol{A}|$ was 82 in this experiment. Again, we used the *conjugate gradient* method to find the local maximum value with the initial value being set to be the parameters obtained in the CRF learning for the *discarded* setting.

Table 5 shows the average performance of POS tagging over 5 different POS unique data. Since the POS ambiguous sentences are only a fraction of all of the training data, the overall performance (P) was slightly improved by the proposed method. However, according to the performance for ambiguously annotated words (APA), the proposed method outperformed other heuristics for POS disambiguation. The P and APA scores between the proposed method and the comparable methods are significantly different based on the *Wilcoxon signed rank test* at the 5% significance level. Although the performance improvement in this POS tagging task was moderate, we believe the proposed method will be more effective to the NLP tasks whose corpus has a considerable number of ambiguous annotations.

6 Related Work

Pereira and Schabes (1992) proposed a grammar acquisition method for partially bracketed corpus. Their work can be considered a generative model for the tree structure output problem using partial annotations. Our discriminative model can be extended to such parsing tasks.

[10] Although the order in which the candidate tags appear has not been standardized in the PTB corpus, we assume that annotators might order the candidate tags with their confidence.

		mrg	random	first	frequent	discarded
Ex.1	P	**94.39**	94.27	94.26	94.27	94.19
	APA	**73.10**	71.58	72.65	71.68	71.91
Ex.2	P	**95.08**	94.98	94.97	94.97	94.98
	APA	**76.70**	74.27	75.28	74.32	75.16

Table 5: The average POS tagging performance over 5 trials.

Our model is interpreted as one of the CRFs with hidden variables (Quattoni et al., 2004). There are previous work which handles hidden variables in discriminative parsers (Clark and Curran, 2006; Petrov and Klein, 2008). In their methods, the objective functions are also formulated as same as equation (3).

For interactive annotation, Culotta et al. (2006) proposed corrective feedback that effectively reduces user operations utilizing partial annotations. Although they assume that the users correct entire label structures so that the CRFs are trained as usual, our proposed method extends their system when the users cannot annotate all of the labels in a sentence.

7 Conclusions and Future Work

We are proposing a parameter estimation method for CRFs incorporating partial or ambiguous annotations of structured data. The empirical results suggest that the proposed method reduces the domain adaptation costs, and improves the prediction performance for the linguistic phenomena that are sometimes difficult for people to label.

The proposed method is applicable to other structured output tasks in NLP, such as syntactic parsing, information extraction, and so on. However, there are some NLP tasks, such as the word alignment task (Taskar et al., 2005), in which it is not possible to efficiently calculate the sum score of all of the possible label configurations. Recently, Verbeek and Triggs (2008) independently proposed a parameter estimation method for CRFs using partially labeled images. Although the objective function in their formulation is equivalent to equation (3), they used *Loopy Belief Propagation* to approximate the sum score for their application (scene segmentation). Their results imply these approximation methods can be used for such applications that cannot use dynamic programming techniques.

Acknowledgments

We would like to thank the anonymous reviewers for their comments. We also thank Noah Smith, Ryu Iida, Masayuki Asahara, and the members of the T-PRIMAL group for many helpful discussions.

References

Anderson, Brigham, Sajid Siddiqi, and Andrew Moore. 2006. Sequence selection for active learning. Technical Report CMU-IR-TR-06-16, Carnegie Mellon University.

Beers, Mark H. 2004. *The Merck Manual of Medical Information (in Japanese)*. Nikkei Business Publications, Inc, Home edition.

Clark, Stephen and James R. Curran. 2006. Partial training for a lexicalized-grammar parser. In *Proceedings of the Annual Meeting of the North American Association for Computational Linguistics*, pages 144–151.

Culotta, Aron, Trausti Kristjansson, Andrew McCallum, and Paul Viola. 2006. Corrective feedback and persistent learning for information extraction. *Artificial Intelligence Journal*, 170:1101–1122.

Keene, Donald, Hiroyoshi Hatori, Haruko Yamada, and Shouko Irabu, editors. 1992. *Japanese-English Sentence Equivalents (in Japanese)*. Asahi Press, Electronic book edition.

Kudo, Taku, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of Empirical Methods in Natural Language Processing*.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*.

Luo, Xiaoquan. 2003. A maximum entropy chinese character-based parser. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 192–199.

Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2).

Mori, Shinsuke. 2006. Language model adaptation with a word list and a raw corpus. In *Proceedings of the 9th International Conference on Spoken Language Processing*.

Peng, Fuchun, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the International Conference on Computational Linguistics*.

Pereira, Fernando C. N. and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of Annual Meeting Association of Computational Linguistics*, pages 128–135.

Petrov, Slav and Dan Klein. 2008. Discriminative log-linear grammars with latent variables. In *Advances in Neural Information Processing Systems*, pages 1153–1160, Cambridge, MA. MIT Press.

Quattoni, Ariadna, Michael Collins, and Trevor Darrell. 2004. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems*.

Sha, Fei and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology-NAACL*, Edmonton, Canada.

Taskar, Ben, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Verbeek, Jakob and Bill Triggs. 2008. Scene segmentation with CRFs learned from partially labeled images. In *Advances in Neural Information Processing Systems*, pages 1553–1560, Cambridge, MA. MIT Press.

Appendix A Computation of Objective and Derivative functions

Here we explain the effective computation procedure for equation (3) and (4) using dynamic programming techniques.

Under the first-order Markov assumption[11], two types of features are usually used: one is pairs of an observed variable and a label variable (denoted as $\boldsymbol{f}(x_t, y_t) : X \times Y$), the other is pairs of two label variables (denoted as $\boldsymbol{g}(y_{t-1}, y_t) : Y \times Y$) at time t. Then the feature vector can be decomposed as $\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{t=1}^{T+1} \boldsymbol{\phi}(x_t, y_{t-1}, y_t)$ where $\boldsymbol{\phi}(x_t, y_{t-1}, y_t) = \boldsymbol{f}(x_t, y_t) + \boldsymbol{g}(y_{t-1}, y_t)$. In addition, let S and E be special label variables to encode the beginning and ending of a sequence, respectively. We define $\boldsymbol{\phi}(x_t, y_{t-1}, y_t)$ to be $\boldsymbol{\phi}(x_t, S, y_t)$ at the head $t = 1$ and $\boldsymbol{g}(y_{t-1}, E)$ at the tail where $t = T + 1$. The technique of the effective calculation of the normalization value is the precomputation of the $\alpha_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{L}}[t, j]$, and$\beta_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{L}}[t, j]$ matrices with given $\boldsymbol{\theta}, \boldsymbol{x}$, and $\boldsymbol{L}$. The matrices α and β are defined as follows, and should be calculated in the order of $t = 1, \cdots, T$, and $t = T + 1, \cdots, 1$, respectively

$$\alpha_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{L}}[t, j] = \begin{cases} 0 & \text{if } j \notin L_t \\ \boldsymbol{\theta} \cdot \boldsymbol{\phi}(x_t, S, j) & \text{else if } t = 1 \\ \ln \sum_{i \in L_{t-1}} e^{\alpha[t-1,i] + \boldsymbol{\theta} \cdot \boldsymbol{\phi}(x_t, i, j)} & \text{else} \end{cases}$$

$$\beta_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{L}}[t, j] = \begin{cases} 0 & \text{if } j \notin L_t \\ \boldsymbol{\theta} \cdot \boldsymbol{g}(j, E) & \text{else if } t = T + 1 \\ \ln \sum_{k \in L_{t+1}} e^{\boldsymbol{\theta} \cdot \boldsymbol{\phi}(x_t, j, k) + \beta[t+1,k]} & \text{else} \end{cases}$$

Note that $\boldsymbol{L} = (Y, \cdots, Y)$ is used to calculate all the entries in $\boldsymbol{Y}$. In the rest of this section, we omit the subscripts $\boldsymbol{\theta}, \boldsymbol{x}$, and $\boldsymbol{L}$ of α, β, Z unless misunderstandings could occur. The time complexity of the $\alpha[t, j]$ or $\beta[t, j]$ computation is $O(T|Y|^2)$.

Finally, equations (3) and (4) are efficiently calculated using α, β. The logarithm of Z in equation (3) is calculated as:

$$\ln Z_{\boldsymbol{\theta},\boldsymbol{Y_L}} = \ln \sum_{j \in L_T} e^{\alpha_{\boldsymbol{\theta},\boldsymbol{L}}[T,j] + \boldsymbol{\theta} \cdot \boldsymbol{g}(j,E)}.$$

Similarly, the first and second terms of equation (4) can be computed as:

$$\sum_{\boldsymbol{y} \in \boldsymbol{Y_L}} P_{\boldsymbol{\theta},\boldsymbol{L}}(\boldsymbol{y}|\boldsymbol{x}) \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in L_T} \varepsilon_{\boldsymbol{L}}(T, i, E) \boldsymbol{g}(i, E) + \sum_{t=1}^{T} \sum_{j \in L_t} \left(\gamma_{\boldsymbol{L}}(t, j) \boldsymbol{f}(x_t, j) + \sum_{i \in L_{t-1}} \varepsilon_{\boldsymbol{L}}(t, i, j) \boldsymbol{g}(i, j) \right)$$

where $\boldsymbol{\theta}, \boldsymbol{x}$ are omitted in this equation, and $\gamma_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{L}}$ and $\varepsilon_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{L}}$ are the marginal probabilities:

$$\begin{aligned} \gamma_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{L}}(t, j) &= P_{\boldsymbol{\theta},\boldsymbol{L}}(y_t = j|\boldsymbol{x}) \\ &= e^{\alpha[t,j] + \beta[t,j] - \ln Z_{\boldsymbol{Y_L}}}, \text{ and} \\ \varepsilon_{\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{L}}(t, i, j) &= P_{\boldsymbol{\theta},\boldsymbol{L}}(y_{t-1} = i, y_t = j|\boldsymbol{x}) \\ &= e^{\alpha[t-1,i] + \boldsymbol{\theta} \cdot \boldsymbol{\phi}(x_t,i,j) + \beta[t,j] - \ln Z_{\boldsymbol{Y_L}}}. \end{aligned}$$

Note that $\boldsymbol{Y_L}$ is replaced with $\boldsymbol{Y}$ and $\boldsymbol{L} = (Y, \cdots, Y)$ to compute the second term.

[11] Note that, although the rest of the explanation based on the first-order Markov models for purposes of illustration, the following arguments are easily extended to the higher order Markov CRFs and semi-Markov CRFs.