Japanese Dependency Analysis Based on SVMs and CRFs

Huiwei Zhou, Tong Yu, Degen Huang

***Abstract*—**This paper presents a method of Japanese dependency structure analysis based on Support Vector Machines (SVMs) and Conditional Random Fields (CRFs). Cascaded chunking model based on SVMs has been proposed and has achieved high accuracy. It parses a sentence deterministically only deciding whether the current segment modifies the segment on its immediate right hand side based on SVMs. We present a method of Japanese dependency structure analysis based on CRFs. We consider Japanese dependency structure analysis as a sequential labeling problem and apply CRFs to label whether each segment modifies the segment on its immediate right hand side. Furthermore, we combine SVMs and CRFs to improve the accuracy of Japanese dependency analyzer. Experiments using the Kyoto University Corpus show that the presented method outperforms previous systems.

***Keywords*—**Conditional Random Fields (CRFs), Japanese dependency analysis, sequential labeling, Support Vector Machines (SVMs)

I. INTRODUCTION

Dependency analysis has been recognized as a basic process in Japanese sentence analysis. And a number of studies have been proposed. Japanese dependency is usually in terms of relationship between phrasal units called bunsetsu segments (hereafter segments).

In recent years, as large-scale tagged corpora have become available, a number of statistical parsing techniques using such tagged corpora have been developed [1] [2] [3] [4]. The previous dependency analysis is divided into two approaches. One approach is based on a statistical model [1] [2] [3]. These models need to calculate the probabilities for all possible dependencies in a sentence to obtain the optimal set of dependency. It is not efficient. The other approach is a cascaded chunking model [4] based on SVMs [5]. The method is simple and efficient. It achieves high accuracy. It parses a sentence deterministically only deciding whether the current segment modifies the segment on its immediate right hand side and it applies SVMs to classify all possible pairs of segments into positive (dependent) or negative (non-dependent) examples.

Conditional random fields (CRFs) [6] are discriminative models applied to sequential labeling problems. CRFs can discriminate the correct sequence from all other candidate sequences without making independence assumption for features. They are considered to be the state-of-the-art framework to date. Empirical successes with CRFs have been reported recently in part-of-speech tagging [6], shallow parsing [7], named entity recognition [8], Chinese word segmentation [9], and Information Extraction [10] [11].

In this paper, we propose an application of CRFs to Japanese dependency structure analysis. We consider Japanese dependency structure analysis as a sequential labeling problem and apply CRFs to label whether each segment modifies the segment on its immediate right hand side. Moreover, we combine SVMs and CRFs to improve the performance of Japanese dependency analyzer.

II. SVMS AND CRFS

A. Support Vector Machine (SVM)

Support Vector Machine (SVM) [5] is one of the binary linear classifiers introduced by Vipnik. Suppose l training examples $(\mathbf{x}_i, y_i)$, $(1 \le i \le l)$ are given, where x_i is a feature vector in n dimensional feature space, y_i is the class label {+1, -1} (positive or negative) of x_i. SVM finds a hyperplane $(\mathbf{w} \cdot \mathbf{x} + b) = 0$ which separate the training examples and has maximum margin between two hyperplane $(\mathbf{w} \cdot \mathbf{x} + b) \ge 1$ and $(\mathbf{w} \cdot \mathbf{x} + b) \le -1$. The optimal hyperplane with maximum margin can be found by solving the following quadratic programming problem.

Manuscript received December 31, 2006; revised March 11, 2007. This work was elaborated in Natural Language Processing Laboratory, Department of Computer Science and Engineering, Dalian University of Technology in CHINA.

Huiwei Zhou received the engineering degree in School of Electrical Engineering from Xi'an Jiaotong University in CHINA, master thesis degree from Graduate School of Information System from the University of Electro Communications in JAPAN. Actually she is a lecturer in Department of Computer Science and Engineering, Dalian University of Technology in CHINA. Her scope of research are Natural Language Processing, Machine Learning and Machine Translation(corresponding author to provide phone: 0086-411-84708140; fax: 0086-411-84706706; e-mail: zhou_huiwei@163.com).

Tong Yu is a master in Department of Computer Science and Engineering, Dalian University of Technology in CHINA. His scope of research are Natural Language Processing, Machine Learning and Machine Translation (e-mail: yttjdl@yahoo.com.cn).

Degen Huang received the engineering and PhD degrees in Department of Computer Science and Engineering from Dalian University of Technology in CHINA. Actually he is a professor in Department of Computer Science and Engineering, Dalian University of Technology in CHINA. His scope of research are Natural Language Processing, Machine Learning, Data Mining and Machine Translation (e-mail: huangdg@dlut.edu.cn).

$$\min \quad \frac{1}{2}\|\mathrm{w}\|^2 \tag{1}$$

$$\text{subject to} \quad y_i(w \cdot x_i + b) \geq 0, \quad i = 1,2,\ldots, l$$

The decision function can be written as:

$$f(x) = \mathrm{sgn}[\sum_{\mathbf{x}_i \in sv} \alpha_i y_i (x_\mathbf{i} \cdot x) + b] \tag{2}$$

Where α_i is the Lagrange multiplier corresponding to each constraint. The Kernel function $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ can reduce the computational overhead when the training example x is projected onto a high dimensional space by using projection function ϕ. Among the many kinds of Kernel functions, the *d-th* polynomial kernel: $K(x_i, y_j) = (x_i \cdot x_j + 1)^d$ is used. Where d is the dimension of the polynomial functions.

Further more, the optimization problem can be written into the following maximum problem.

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \tag{3}$$

Finally, the label of an unknown example is decided by the following function:

$$f(x) = \mathrm{sgn}[\sum_{\mathbf{x}_i \in sv} \alpha_i y_i K(x_i \cdot x) + b] \tag{4}$$

SVM estimate the label of an unknown example whether sign of $f(x)$ is positive(+1) or negative(-1).

B. Conditional Random Fields (CRFs)

Conditional random fields (CRFs) [6] are undirected graphical models trained to maximize a conditional probability. In the special case the graph structure is a linear chain, which corresponds to a finite state machine, and is suitable for sequence labeling. Let $X= (X_1, X_2, \ldots,X_n)$ be some observed input data sequence, such as a sequence of words in a sentence. Let $Y= (Y_1, Y_2, \ldots,Y_n)$ be some sequence of states. CRFs define the conditional probability of a state sequence given an input sequence as

$$P(Y \mid X) = \frac{1}{Z_X}\exp(\sum_{i=1}^{n}\sum_{k}\lambda_k f_k(Y_{i-1}, Y_i, X, i)) \tag{5}$$

Where Zx is a normalization factor over all state sequences,

$$Z_X = \sum_{Y \in Y^N}\exp(\sum_{i=1}^{n}\sum_{k}\lambda_k f_k(Y_{i-1}, Y_i, X, i)) \tag{6}$$

Z_X is the sum of the "scores" of all possible state sequences. $f_k(Y_{i-1}, Y_i, X, t)$ is an arbitrary feature function over its arguments, and λ_k is a learned weight for each feature function f_k. The feature functions can measure any aspect of a state transition, $Y_{i-1} \rightarrow Y_i$, and the entire observation sequence, x, centered at the current time step, t.

The most probable label sequence for an input sequence X is then given by

$$\hat{Y} = \arg\max_{Y \in Y^N} P(Y \mid X) = \arg\max \Lambda \bullet F(Y, X) \tag{7}$$

which can be determined using the Viterbi algorithm.

CRFs are trained using the maximum likelihood—maximizing the conditional probability of a set of label sequences. The log-likelihood of training set $T=\{<X_i,Y_i>i=1,\ldots,N\}$ is written

$$\begin{aligned} L_\Lambda &= \sum_i \log P_\Lambda(Y_i \mid X_i) \\ &= \sum_i \log(\sum_{Y \in Y^N}\exp(\Lambda \bullet [F(Y_i, X_i) - F(Y, X_i)])) \\ &= \sum_i [\Lambda \bullet F(Y_i, X_i) - \log(Z_{X_i})] \end{aligned} \tag{8}$$

CRFs can be trained by traditional iterative scaling algorithms, such as GIS and IIS [12] or quasi-Newton methods [13].

III. Japanese Dependency Analysis Based on SVMs and CRFs

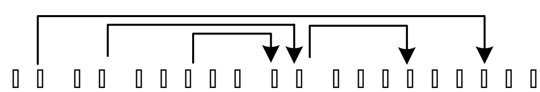
A. Cascaded Chunking Model

We define a sentence as a sequence of segments $B = \langle b_1, b_2, \ldots, b_m \rangle$ and its syntactic structures as a sequence of dependency patterns $D = \langle dep(1), dep(2), \ldots, dep(m-1) \rangle$, where $dep(i) = j$ means that segment b_i depends on (modifies) segment b_j. In this frame-work, we suppose that the dependency sequence satisfies the following constrains.

1. Except for the rightmost one, each segment depends on exactly one of the segments appearing to the right.
2. Dependencies do not cross each other.

Cascaded chunking model has been applied to Japanese dependency analysis [4]. Japanese dependency analysis using cascaded chunking is as follows:

1. Put an O tag on all segments since the dependency relation of each one is undecided.



(Mother bought many thing for me.)

母は　私に　いろいろな 物を　買ってくれました。
Mother　me　many　thing　buy
Initialization
Input: 母は　私に　いろいろな　物を　買ってくれました。
Tag: O　O　O　O　O

Input: 母は　私に　いろいろな　物を　買ってくれました。
Tag: O　O　D(Del.)　D　O

Input: 母は　私に　物を　買ってくれました。
Tag: O　D(Del.)　D　O

Input: 母は　物を　買ってくれました。
Tag: O　D(Del.)　O

Input: 母は　買ってくれました。
Tag: D(Del.)　O

Input: 買ってくれました。
O(Finish)

Fig. 1. Example of the parsing process with cascaded chunking model

2. For each segment with an O tag, decide whether it modifies the segment on its immediate right hand side. If so, the O tag is replaced with a D tag.
3. Delete all segments with a D tag that are immediately followed by a segment with an O tag.
4. Terminate the algorithms if a single segment remains, otherwise return to step 2 and repeat.

Fig. 1 shows an example of the parsing process and the result. It is simple and efficient. It parses a sentence deterministically only deciding whether the current segment modifies the segment on its immediate right hand side. Taku Kuto used SVMs to determine whether a pair of segments is in a dependency relation or not because of their state-of-the-art performance and generalization ability.

In this paper, we propose a method combining SVMs with CRFs to determine whether a pair of segments is in a dependency relation or not.

B. Cascaded Chunking Model Based on SVMs

In order to use SVMs for dependency analysis, we adopt a sample method: we take a pair of segments that are in a dependency relation as a positive data, and a pair of segments that are not in a dependency relation as a negative data.

In training, the model simulated the parsing algorithm by consulting the correct answer from the training annotated corpus. In testing, the model consults the trained system and parses the input sentence with the parsing algorithm.

TABLE I
FEATURES USED IN SVMS

Static Features	left/right segments	Head Word (surface-form, POS, POS-subcategory, inflection-type, inflection-form), Functional Word (surface-form, POS, POS-subcategory, inflection-type, inflection-form), brackets, quotation-marks, punctuation-marks, position in sentence (beginning, end)
	Between two segments	Distance (1,2-5,6-), case-particles, brackets, quotation-marks, punctuation-marks
Dynamic Features	The segments which modify the current candidate modifee or modifer	Form of inflection represented with Functional Representation
	The segment which is modified by the current candidate modifee	POS and POS-subcategory of Head word

The features used in SVMs are shown in Table I. The features include static features and the dynamic features.

C. Cascaded Chunking Model Based on CRFs

In order to use CRFs for dependency analysis, we cast Japanese dependency structure analysis problem as one of sequence tagging: the segments that modify the segments on their immediate right hand side are given the D tag, otherwise are given the O tag. The task of determining whether the current segment modifies the segment on its immediate right hand side becomes a matter of assigning a sequence of tags to the input sequence of Japanese sentence.

And then delete all segments with a D tag that are immediately followed by a segment with an O tag according cascaded chunking model. Terminate the algorithms if a single segment remains, otherwise repeat to assign a sequence of tags to the remained sequence of the sentence.

TABLE II
FEATURE TEMPLETES USED IN CRFS

Unigram basic features	$<hp1_i>,<hp2_i>, <hcf_i>,<hct_i>, <hbw_i>, <fp1_i>,<fp2_i>, <fcf_i>,<fct_i>, <fbw_i>, <hp1_j>,<hp2_j>, <hcf_j>,<hct_j>, <hbw_j>, <fp1_j>,<fp2_j>, <fcf_j>,<fct_j>, <fbw_j>$, <*brackets*$_i$>,<*brackets*$_j$>, <*quotation-marks*$_i$>, <*quotation-marks*$_j$>, <*punctuation-marks*$_i$>, <*punctuation-marks*$_j$>, <*position of the segment i in sentence (beginning, end)*>, <*position of the segment j in sentence (beginning, end)*>, <*case-particles*$_i$>,<*case-particles*$_j$>	
Dynamic Features	The segments which modify the segment *i* and *j*	<*Form of inflection represented with Functional Representation*>
	The segment which is modified by the segment *j*	<*POS of head word*>, <*POS-subcategory of head word*>
Bigram basic features	$<hp1_i , hp1_j >,<hp2_i , hp2_j >$, $<hcf_i , hcf_j >,<hct_i , hct_j >$, $<hbw_i , hbw_j >,<fp1_i , fp1_j >$, $<fp2_i , fp2_j >,<fcf_i , fcf_j >,<fct_i , fct_j >$, $<fbw_i , fbw_j >$, $<hbw_i , hp1_i , hbw_j\ hp1_j >$, $<fbw_i , fp1_i , fbw_j ,fp1_j >$, $<hp1_i , hp2_i , hp1_j\ hp2_j >$, $<hbw_i , hp1_i , hp2_i , hbw_j ,hp1_j ,hp2_j >$, $<fbw_i , fp1_i , fp2_i , fbw_j ,fp1_j ,fp2_j >$, $<hbw_i , hp1_i , hp2_i , hcf_i , hct_i , hp1_j ,hp2_j >$, $<fbw_i , fp1_i , fp2_i , fcf_i , fct_i , fp1_j ,fp2_j >$, $< hp1_i , hp2_i , hbw_j ,hp1_j ,hp2_j ,hcf_j , hct_j >$, $< fp1_i , fp2_i , fbw_j ,fp1_j ,fp2_j ,fcf_j , fct_j >$, $< hbw_i , hp1_i , hp2_i , hcf_i ,hct_i , hbw_j ,hp1_j ,hp2_j ,hcf_j , hct_j >$, $< fbw_i , fp1_i , fp2_i , fcf_i ,fct_i , fbw_j ,fp1_j ,fp2_j ,fcf_j , fct_j >$,	

$$f_k(Y_{i-1}, Y_i, X, i)$$

$$x_i =< hp1_i, hp2_i, hcf_i, hct_i, hbw_i, fp1_i, fp2_i, fcf_i, fct_i, fbw_i >$$

$$x_j =< hp1_j, hp2_j, hcf_j, hct_j, hbw_j, fp1_j, fp2_j, fcf_j, fct_j, fbw_j >$$

where $hp1_i / hp1_j$ and $hp2_i / hp2_j$ are the POS and POS-subcategory of head word, hcf_i / hcf_j and hct_i / hct_j are the inflection-type and inflection-form of head word, hbw_i / hbw_j are the surface-form of head word, $fp1_i / fp1_j$ and $fp2_i / fp2_j$ are the POS and POS-subcategory of functional word, fcf_i / fcf_j and fct_i / fct_j are the inflection-type and inflection-form of functional word, fbw_i / fbw_j are the surface-form of functional word

Feature templates used in CRFs are shown in Table II.

D. Cascaded Chunking Model Based on SVMs and CRFs

We combine SVMs with CRFs to analysis Japanese dependency structure. We apply SVMs and CRFs to assign a sequence of tags to the input sequence of Japanese sentence. If the SVMs based tag is the same with the CRFs based tag for a segment *i*, put that tag on the segment. If the SVMs based tag is different from the CRFs based tag for a segment *i*, we assign the tag of the segment *i* according to the export value of SVMs model and CRFs model.

The separate hyperplane of SVMs classifier is H. The distance $d(d \geq 0)$ from a pair of segments f_{ij} to the separate hyperplane H is defined as:

$$d = \left| \sum_{k,l;\, f_{kl} \in SVs} \alpha_{kl} y_{kl} K(f_{kl} \cdot f_{ij}) + b \right| \quad (9)$$

The tagging probability *P(i)* of segment *i* based on CRFs is defined as:

$$P(i) = \frac{1}{Z_X} \exp\left(\sum_k \lambda_k f_k(Y_{i-1}, Y_i, X, i) \right) \quad (10)$$

We assign the tag of the segment *i* according to the distance *d* of SVMs model and the tagging probability *P(i) of* CRFs model. There are four conditions:

(1) $d < \varepsilon_{SVM}$ and $P(i) < \varepsilon_{CRF}$

We consider the SVMs based tag and the CRFs based tag are both unbelievable .

(2) $d > \varepsilon_{SVM}$ and $P(i) < \varepsilon_{CRF}$

We consider the SVMs based tag is believable and the CRFs based tag is unbelievable .

(3) $d < \varepsilon_{SVM}$ and $P(i) > \varepsilon_{CRF}$

We consider the SVMs based tag is unbelievable and the CRFs based tag is believable .

(4) $d > \varepsilon_{SVM}$ and $P(i) > \varepsilon_{CRF}$

We consider the SVMs based tag and the CRFs based tag are both believable .

For condition (2), put the SVMs based tag on the segment *i*. For condition (3), put the CRFs based tag on the segment *i*. For condition (1) and (4), we put the SVMs based tag on the segment *i*.

We can control the dependency accuracy by adjust the threshold $\varepsilon_{SVM} (0 < \varepsilon_{SVM} < 1)$ and $\varepsilon_{CRF} (0 < \varepsilon_{CRF} < 1)$.

IV. Experiments and Discussion

A. Experiments Setting

We use Kyoto University text corpus (Version 3.0) consisting of articles of Mainichi Newspaper. The sentences from the articles on January 1st, 3rd to 9th are used for the training data, and the sentences from the articles on January 10th are used for the test data. Our experiments are under the condition $d = 3$ (dimension of the polynomial functions used for the Kernel function).

B. Experimental Results

The experimental results are shown in Table III. Table III shows the method based on SVMs and CRFs outperforms the cascaded chunking model based on sole SVMs [14] and sole CRFs. We have proposed an improved SVMs-NN-LSVM [14] to increase the dependency accuracy. NN-LSVM pruned those samples that unused or not good to improve the classifier's performance. In this paper, we also pruned the training samples using NN-LSVM. The CRFs based method is not as good as the SVMs based method.

TABLE III
Results based on NN-LSVM

Mothed	Dep. Acc. (%)	Sen. Acc.(%)
Sole SVMs	89.86	49.14
Sole CRFs	87.31	42.24
Combining SVMs and CRfs	90.03	49.29

Table IV shows that the approach combining SVMs and CRFs achieved higher dependency accuracy and sentence accuracy than the sole SVMs based method when $\varepsilon_{SVM} < 0.4$. This means that the CRFs based method performs better than the SVMs based method near the separate hyperplane of SVMs classifier.

TABLE IV
Results based on SVMs and CRFs

ε_{SVM}	ε_{CRF}	Dep. Acc. (%)	Sen. Acc. (%)
0.1	0.1	90.03	49.29
0.1	0.3	90.03	49.29
0.1	0.5	90.01	49.29
0.1	0.7	90.00	49.29
0.1	0.9	89.99	49.29
0.2	0.1	89.94	49.29
0.2	0.3	89.94	49.29
0.2	0.5	89.92	49.29
0.2	0.7	89.96	49.29
0.2	0.9	89.94	48.89
0.3	0.1	89.86	48.56
0.3	0.3	89.86	48.56
0.3	0.5	89.86	48.56
0.3	0.7	89.97	49.16
0.3	0.9	89.96	49.03
0.4	0.1	89.80	48.29
0.4	0.3	89.80	48.29
0.4	0.5	89.80	48.29
0.5	0.1	89.57	47.82
0.5	0.3	89.57	47.82
0.5	0.5	89.57	47.88

However, the approach combining SVMs and CRFs cannot performance better than the sole SVMs when $\varepsilon \geq 0.4$.

TABLE V
Comparison with the related work

Model	Training Corpus (# of days)	Dependenc y Acc. (%)	Sentence Acc. (%)
Combining SVMs and CRFs	Kyoto Univ. (8)	90.03	49.29
Improved SVM[14]	Kyoto Univ. (8)	89.86	49.14
Cascaded chunking[4]	Kyoto Univ. (8)	89.29	47.53
Probabilistic (ME)[1]	Kyoto Univ. (8)	87.14	40.60
Probabilistic (ME + posterior context)[2]	Kyoto Univ. (8)	87.93	43.58

C. Comparison with Related Work

The results of our model and the recent Japanese Dependency Analysis model (cascaded chunking [3], ME [1], ME + posterior context [2]) are summarized in Table V. Dependency accuracy and sentence accuracy are improved combining SVMs and CRFs.

V. CONCLUSION

This paper proposed an application of CRFs to Japanese dependency structure analysis. We consider Japanese dependency structure analysis as a sequential labeling problem and apply CRFs to label whether each segment modifies the segment on its immediate right hand side. Furthermore, this paper presented a method for Japanese dependency analysis that combining SVMs and CRFs. Experiments show that the approach combining SVMs and CRFs outperforms the sole SVMs based method and the sole CRFs based method. We achieve higher dependency accuracy (90.03%) and sentence accuracy (49.29%) with a small training set. Since the CRFs based method performs better than the SVMs based method near the separate hyperplane of SVMs classifier.

REFERENCES

[1] Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara, "Japanese Dependency Structure Analysis Based on Maximum Entropy Models," In *Proc. of the EACL*, 1999, pp. 196-203.

[2] Kiyotaka Uchimoto, Masaki Murata, Satoshi Sekine, and Hitoshi Isahara, "Dependency Model using posterior context," In *Proc. of Sixth International Workshop on Parsing Technologies*, 2000, pp. 321-322.

[3] Taku Kuto and Yuji Matsumoto, "Japanese Dependency Structure Analysis Based on Support Vector Machines," In *1999 Joint Sigdat Conference on Empirical Methods in Natural Language processing and Very Large Corpora*, 2000, pp. 18-25.

[4] Taku Kuto and Yuji Matsumoto, "Japanese Dependency Analysis using Cascaded Chunking," In *Proc. of CoNLL*, Taipei, Taiwan, 2002, pp. 63-69.

[5] Vapnik, V.N., *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, 1995.

[6] John Lafferty, Andrew McCallum, and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," In *Proc. of ICML*, 2001, pp. 282-289.

[7] Fei Sha and Fernando Pereira, "Shallow parsing with conditional random fields," In *Proc. of HLT-NAACL*, 2001, pp. 213-220.

[8] Andrew McCallum and Wei Li, "Early results for named entity recognition with conditional random fields, feature induction and webenhanced lexicons," In *Proc. of CoNLL*, 2003, pp. 188-191.

[9] Fuchun Peng, Fangfang Feng, and Andrew McCallum, "Chinese segmentation and new word detection using conditional random fields," In *Proc. of COLING*, 2004, pp. 562-568.

[10] Fuchun Peng and Andrew McCallum, "Accurate information extraction from research papers," In *Proc. of HLT/NAACL*, 2004.

[11] David Pinto, Andrew McCallum, Xing Wei, and W. Bruce Croft, "Table extraction using conditional random fields," In *Proc. of SIGIR*, 2003, pp. 235-242.

[12] Della Pietra, Stephen, Vincent J. Della Pietra, and John D. Lafferty, "Inducing features of random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, Vol.19, No.4, pp. 380-393.

[13] Dong C. Liu and Jorge Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, 1989, Ser. B, Vol 45, Iss 3, pp. 503-528.

[14] Huiwei Zhou, Wei Li, and Degen Huang, "Japanese Dependency Analysis Using NN-LSVM," *JSCL-2005*, Nanking, 2005, pp. 295-301.