Improving Named Entity Extraction Accuracy using Unlabeled Data and Several Extractors

Tomoya Iwakura and Seishi Okamoto

***Abstract*—This paper proposes feature augmentation methods using unlabeled data and several Named Entity (NE) extractors. We collect NE-related information of each word (which we call NE-related labels) from unlabeled data by using NE extractors. NE-related labels which we collect include candidate NE class labels of each word and NE class labels of co-occurring words. To accurately collect the NE-related labels from unlabeled data, we consider methods to collect NE-related labels by using outputs of several NE extractors. We use NE-related labels as additional features for creating new NE extractors. We apply our NE extraction methods using the NE-related labels to IREX Japanese NE extraction task. The experimental results show better accuracy than the previous results obtained with NE extractors using handcrafted resources.**

***Index Terms*—Named entity recognition, unlabeled data, combination of extractors.**

I. Introduction

NAMED Entity (NE) extraction is one of the basic technologies used in text processing like information extraction and question answering. NE extraction aims to extract proper nouns and numerical expressions in text, such as persons, locations, organizations, dates, times, and so on.

To implement NE extractors, the following approaches are mainly used. The first approach is handcrafted-rule based NE extractions [1]. The others are machine learning ones, such as unsupervised learning, semi-supervised learning and supervised learning.

In those machine leaning based methods, supervised learning is widely researched recently [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. Furthermore, supervised learning based NE extractors have shown state-of-the-art performance [5], [6], [7], [9].

In the supervised learning based approaches, information obtained with handcrafted lexical resources are often used as features for creating more accurate NE extractors. The handcrafted lexical resources include gazetteers, proper noun dictionaries, thesaurus, and so on. Experimental results obtained with NE extractors using such handcrafted resources have shown better performances than results obtained with NE extractors without using them [3], [5], [6], [7], [8], [9]. However, the construction of the lexical resources is very time-consuming.

Manuscript received November 4, 2008. Manuscript accepted for publication August 25, 2009.

Tomoya Iwakura and Seishi Okamoto are with Fujitsu Laboratories Ltd., 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki 211-8588, Japan ({iwakura.tomoya,seishi}@jp.fujitsu.com)

To avoid or alleviate the construction of training data and external lexical resources, a number of approaches that exploit unlabeled data have been proposed. For example, various bootstrapping methods for finding rules and dictionaries from unlabeled data have been proposed [13], [2], [14].

Another approach is semi-supervised learning, which uses labeled and unlabeled data for training. For example, the following approaches have been proposed and shown to improve performance; co-training [15] automatically bootstraps labels, Expectation Maximization (EM) which has a theoretical base of likelihood maximization of incomplete data, and structural learning [12] which seeks shared predictive structures jointly learning multiple classification problems.

Feature augmentation methods using unlabeled data have been also applied to Natural Language Processing (NLP) tasks [16], [10], [11]. These techniques use word clusters created from unlabeled data by clustering algorithms as features. These experimental results by using word clusters as features have shown improving performance of supervised learning based NE extractors [10], [11].

This paper proposes feature augmentation methods for NE extractions. Compared to the previous works, our methods use candidate NE classes of words and candidate class labels of co-occurring words (NE-related labels which we call) as features. We collect these NE-related labels from unlabeled data. Furthermore, to collect NE-related labels accurately, our methods use outputs of several NE extractors. We use NE-related labels as new features for creating NE extractors.

This paper is organized as follows. In section II, we describe our Japanese NE extraction method. We describe our collection methods of NE-related labels in section III and report the experimental results on the IREX [17] Japanese NE task in section IV. Finally, we conclude the paper in section V.

II. Japanese Named Entity Extraction

A problem of Japanese NE extraction is that each Japanese NE consists of one or more words, or includes a substring of a word. In this section, we present the chunk representations for word chunks that become NEs at the first. Next, we describe our Japanese NE extraction methods by combining the following methods; NE extraction by word-unit chunking for extracting each NE that consists one or more words, and NE extraction by character-unit chunking for extracting each NE that includes a substring of a word. Finally, we describe

TABLE I
NE EXAMPLES DEFINED BY IREX COMMITTEE

ARTIFACT	LOCATION	ORGANIZATION	PERSON
Nobel Prize in Chemistry	Japan	the Ministry of Foreign Affairs	Tarou Yamada

DATE	MONEY	PERCENT	TIME
May	100 JPY	100%	10:00 a.m.

	田中 *(Tanaka)*	使節 *(mission)*	団 *(party)*	は *(particle)*	日 *(Japan)*	米 *(U.S.A)*	間 *(between)*
IOB1	I-ORG	I-ORG	I-ORG	O	I-LOC	B-LOC	O
IOB2	B-ORG	I-ORG	I-ORG	O	B-LOC	B-LOC	O
IOE1	I-ORG	I-ORG	I-ORG	O	E-LOC	I-LOC	O
IOE2	I-ORG	I-ORG	E-ORG	O	E-LOC	E-LOC	O
SE	B-ORG	I-ORG	E-ORG	O	S-LOC	S-LOC	O

Tanaka mission party dose ... between Japan and U.S.A.

Fig. 1. Examples of NE labels

Support Vector Machines (SVMs) that is the machine learning algorithm for crating NE extractors in our experiments.

A. Chunk Representation for Named Entity Extraction

We use IREX Japanese NE extraction task [17] to evaluate our methods. Table I lists the eight NE classes defined by IREX committee. One of the problems for extracting NEs is that each NE consists of one or more words. To extract NEs, we have to identify word chunks with their NE classes.

To identify word chunks, we use methods to annotate chunk tags to words. We use the following five chunk tag sets; IOB1 [18], IOB2, IOE1, IOE2 [19] and Start/End (SE) [3].

- IOB1: This representation uses three tags which are I, O and B, to represent the inside, outside and beginning of a chunk. B is only used at the beginning of a chunk which immediately follows another chunk. O is used for the outside of any chunk in all the chunk representations.
- IOB2: This representation uses the same three tags outlined for IOB1. However, B tag has a different meaning. B tag is given for the word at the beginning of a chunk.
- IOE1: This representation uses three tags which I, O and E, to represent the inside, outside and end of a chunk. E tag is used to mark the last word of a chunk immediately preceding another chunk.
- IOE2: This representation uses the same three tags outlined for IOE1. However, E tag has a different meaning. E tag is given for the word at the end of a chunk.
- SE: This representation uses five tags which are S, B, I, E and O, for representing chunks. S means that the current word is a chunk consisting of only one word. B means the start of a chunk consisting of more than one word. E means the end of a chunk consisting of more than one word. I means the inside of a chunk consisting of more than two words. O means the outside of any chunk.

We defined five NE label sets for IREX NE extraction tasks by using these five chunk representations. We use "O" to designate words to which none of the NE categories. Each label set in IOB1, IOB2, IOE1 and IOE2 based label sets has $(8 \times 2) + 1 = 17$ labels, and the SE based NE label set has $(8 \times 4) + 1 = 33$ labels. Figure 1 shows examples for these five representations.

B. Named Entity extraction by word-unit chunking

NE extraction by word-unit chunking identifies word chunks consisting of NEs and classifies word chunks into NE categories. We consider methods to assign one of the NE labels to each word for extracting NEs.

We follow the previous Japanese NE extraction methods that have shown better performance [3], [6], [8], [9] for feature selection. Our NE extractors use the following features of the current word, the preceding two words and the two succeeding words as features (5-word window). This setting has shown the best performance in several Japanese NE experiments [3], [6], [8], [9]. In the following explanations, we assume that a sentence consists of m words $\{w_1, ..., w_m\}$ $(0 < m)$.

- Word: We use words tokenized with a morphological analyzer because Japanese has no word boundary marker. We use ChaSen[1] as the morphological analyzer. When we classify i-th word w_i $(1 \leq i \leq m)$ to one of the NE labels, we use w_{i-2},w_{i-1}, w_i, w_{i+1} and w_{i+2} as features.
- Part of speech (POS) tags: We use POS tags of words assigned by ChaSen. Let $POS(w)$ be the POS tag of a word w. We use $POS(w_{i-2})$, $POS(w_{i-1})$, $POS(w_i)$, $POS(w_{i+1})$ and $POS(w_{i+2})$ as features.

[1] http://chasen.naist.jp/hiki/ChaSen/

TABLE II
BASIC CHARACTER TYPE AND NUMBER TYPE

Character type	Hiragana (Japanese syllabary characters), Katakana, Kanji (Chinese letter) Capital alphabet, Lower alphabet, Others.
Number type	$n \leq 12$, $25 \leq n$, $n \leq 100$, $n \leq 2000$, $n < 2000$

Word	POS	CharType	NE label
田中	Noun-Surname	Kanji+	B-ORG
使節	Noun-General	Kanji+	I-ORG
団	Noun-Suffix-General	Kanji	E-ORG
は	Particle-Case-General	Hiragana	O
訪米	Noun-Verb-Connection	Kanji+	S-LOC

↓

Char	POS	CharType	NE label of word	Word	Word CharType	NE label
田	B-Noun-Surname	Kanji	B-B-ORG	B-田中	Kanji+	B-ORG
中	E-Noun-Surname	Kanji	E-B-ORG	E-田中	Kanji+	I-ORG
使	B-Noun-General	Kanji	B-I-ORG	B-使節	Kanji+	I-ORG
節	E-Noun-General	Kanji	E-I-ORG	E-使節	Kanji+	I-ORG
団	S-Noun-Suffix-General	Kanji	S-E-ORG	S-団	Kanji	I-ORG
は	S-Particle-Case-General	Hiragana	S-O	S-は	Hiragana	O
訪	B-Noun-Verb-Connection	Kanji	B-S-LOC	B-訪米	Kanji	O
米	E-Noun-Verb-Connection	Kanji	E-S-LOC	E-訪米	Kanji	B-LOC

Fig. 2. Feature expression for training: The top table is an example of word unit chunking one and the bottom table is character unit chunking one.

– Character types: If a word consists of only one character, the character type is expressed by using the corresponding character types listed in Table II. If a word consisted of more than one character, the character type is expressed by using a combination of the basic character types listed in Table II, such as Kanji-Hiragana[2]. If a word is number, the character type is expressed by the number-type determined by the number type range listed in Table II.
  Let $CT(w)$ be the character type of a word w. We use $CT(w_{i-2})$, $CT(w_{i-1})$, $CT(w_i)$, $CT(w_{i+1})$ and $CT(w_{i+2})$ as features.
– NE labels of preceding words: We use NE labels assigned to the preceding two words in the extraction direction as features. Kudo and Matsumoto proposed to combine English base phrase parsers by voting [20]. To create several English base phrase parsers from a training corpus, they use distinct chunk representation with two parsing directions (Forward/Backward). We also consider the two parsing directions to create NE extractors. Let be $NEL(w)$ the NE label assigned to a word w. If the parsing direction is the end to the begin of a sentence, we use $NEL(w_{i-2})$ and $NEL(w_{i-1})$ as features. If the parsing direction is the begin to the end of a sentence, we use $NEL(w_{i+1})$ and $NEL(w_{i+2})$ as features. In addition to the two parsing direction, we create NE extractors without using the predicted labels as features.

[2]The same character type sequence expressed more than one time is denoted by the character type and "+". For example, character types of "Yamada" are "Capital-alphabet&Small-alphabet+".

To distinguish the same features appeared within 5-word window, features of each word are expressed with the position from the current word. For example, features of two preceding word from the current position i is expressed like "-2-th-word=w_{i-2}" and "-2-th-POS=$POS(w_{i-2})$".

For each NE label set, the following NE extractors are created: NE extractors start parsing from either the beginning or end of sentence direction, and NE extractors parse sentences without using the predicted labels as features. Finally, we implement $(5 \times 3) = 15$ NE extractors.

C. Named Entity extraction by character-unit chunking

Japanese NEs sometimes include a part of a word becoming beginning or end of an NE. To extract Japanese NEs including a part of a word, we apply a character-unit-chunking method.

For example, the "訪米 (visit U.S.A)" in "田中(Tanaka)使節(mission)団(party)は(particle)訪米(visited U.S.A) (Tanaka mission party visit U.S.A.). " dose not match with LOCATION "米(U.S.A)" because this sentence is tokenized as "田中(Tanaka)/使節(mission) /団(party)/は (particle)/訪米(visited U.S.A)", where "/" indicates a word boundary.

To solve this problem, we use a character-unit-chunking-based NE extraction algorithm [8], [9]. Figure 2 shows the examples of a word-unit-chunk representation in the top and a character-unit-chunk representation in the bottom. 2.

We follow the best model of Asahara and Matsumoto [8] for selecting features and chunk representation of character-unit

TABLE III
EXAMPLES OF EXTRACTION RESULTS (TOP) AND EXAMPLES OF NE-RELATED LABELS COLLECTED FROM THE EXTRACTION RESULTS (BOTTOM)

田中/B-ORG	株式会社/E-ORG	上場/O
(Tanaka)	(Co.Ltd.)	(go public)
田中/S-PERSON	社長/O	
(Tanaka)	(president)	
田中/S-PERSON	さん/O	
(Tanaka)	(Mr.)	

↓

Word	Position from the current word	Candidate NE labels in each position	Freq. of NE labels	Ranking in each position
田中 (Tanaka)	current	S-PERSON	2	1
		B-ORG	1	2
	next	O	2	1
		E-ORG	1	2
	two back	O	1	1
さん (Mr.)	current	O	2	1
	previous	S-PERSON	2	1

chunking. In addition to the current position, we use the two preceding and two succeeding characters (5-character window) in character-unit chunking. In the sequel, we assume that a sentence consists of n characters $\{c_1, ..., c_n\}$ ($0 < n$).

- Characters: We use each character as a feature. When we classify i-th character c_i ($1 \leq i \leq n$), we use c_{i-2},c_{i-1},c_i, c_{i+1} and c_{i+2} as features.
- Character types: We assign one of the character types listed in Table II to each character and use it as a feature. Let $CT(c)$ be the character type of a character c, then we use $CT(c_{i-2})$, $CT(c_{i-1})$,$CT(c_i)$, $CT(c_{i+1})$ and $CT(c_{i+2})$ as features.
- Words including characters: We use the words including characters within a widow size as features. Let be $W(c_i)$ the word including i-th character c_i and $P(c_i)$ be the identifier that indicates the position where c_i appears in $W(c_i)$. We combine $W(c_i)$ and $P(c_i)$ for creating a feature. $P(c_i)$ is one of the followings; begging of a word (B), inside of a word (I), end of a word (E) and a character is a word (S).
  For example, if "外務省*(the Ministry of Foreign Affairs)* は*(particle)*" is segmented as "外務省/は", then words including characters are follows; "W(外) = 外務省", "W(務) = 外務省, "W(省) = 外務省" and "W(は)=は". The identifiers that indicate positions where characters appear are follows; "P(外) =B", "P(務) = I", "P(省) = E" and "P(は)=S".
- POS tags of words including characters: Let be $POS(W(c_i))$ the POS tag of the word $W(c_i)$ including i-th character c_i. We use the POS tags of words including characters within window size as features. We express these features with the position identifier $P(c_i)$.
- Character types of words including characters: Let $CT(W(c_i))$ be the character type of the word including i-th character c_i. We use $CT(W(c_{i-2}))$, $CT(W(c_{i-1}))$, $CT(W(c_i))$, $CT(W(c_{i+1}))$ and $CT(W(c_{i+2}))$ as features.
- NE labels of words assigned by a word-unit NE extractor: Let $NEL(W(c_i))$ be the NE label of the word including i-th character c_i. We express these features with the identifier $P(c_i)$ and NE label $NEL(W(c_i))$. In this experiment, we use "$P(c_{i-2}) - NEL(W(c_{i-2}))$", "$P(c_{i-1}) - NEL(W(c_{i-1}))$", "$P(c_i) - NEL(W(c_i))$", "$P(c_{i+1}) - NEL(W(c_{i+1}))$" and "$P(c_{i+2}) - NEL(W(c_{i+2}))$" as features.
- NE labels of preceding extraction results: The NE labels of two preceding extraction results are used as features in the direction of the end to begin of a sentence. The setting have shown good performance in past experiments [8], [9]. Let $NEL(c)$ be the NE label assigned to character c by an NE extractor. In this experiment, we use $NEL(c_{i+1})$ and $NEL(c_{i+2})$ as features.

To distinguish the same features appeared within 5-character window, features of each character are expressed with the position of character from current character. For example, two preceding character from current position i is expressed like "-2-th-character=c_{i-2}".

Each character is classified into one of the $(8 \times 2 + 1) = 17$ NE labels represented by the IOB2 representation. To use the same character-unit-chunking-based NE extractor, we convert IOB1, IOB2, IOE1 and IOE2 based NE extractors output into the SE representation.

D. Support Vector Machines-based NE extractor

We used the chunker YamCha [21], which is based on Support Vector Machines (SVMs) [22], to implement NE extractors. Below we briefly describe SVMs based NE extraction. Suppose we have a set of training data for a binary class problem: $(x_1, y_1), ..., (x_N, y_N)$, where $x_i \in R^n$ is an n dimension feature vector of the i-th sample in the training data and $y_i \in \{+1, -1\}$ is the label of the sample.

TABLE IV
TRAINING AND EVALUATION DATA FOR JAPANESE NE EXTRACTION IN THIS EXPERIMENT

	Training	Evaluation					
NE / Data	CRL data	formal-run GENERAL	formal-run ARREST	domain-specific training	dry-run	dry-run training	Total
ARTIFACT	871	49	13	11	42	67	182
DATE	3654	277	72	69	110	137	665
LOCATION	5660	416	106	165	192	255	1134
MONEY	390	15	8	19	33	32	107
ORGANIZATION	3813	389	74	80	214	270	1027
PERCENT	500	21	0	3	6	19	49
PERSON	3870	355	97	94	169	138	853
TIME	503	59	19	18	24	8	128
Total	19261	1581	389	459	790	926	4145

TABLE V
NE EXTRACTION RESULT WITHOUT UNLABELED DATA ($F_{\beta=1}$): "-F", "-B", AND "-N" INDICATE FORWARD DIRECTION NE EXTRACTION, BACKWARD DIRECTION NE EXTRACTION, AND NE EXTRACTION WITHOUT USING PRECEDING NE LABELS, RESPECTIVELY. AV.($F_{\beta=1}$) AND AV.RANK ARE FOR FIVE PIECES DATA.

Chunk Rep. / Data	CRL Cross	formal-run GENERAL	formal-run ARREST	domain-specific training	dry-run	dry-run training	Av. ($F_{\beta=1}$)	Av. rank
IOB1-F	82.97	84.17	86.02	88.94	81.32	82.10	83.89	8.6
IOB1-B	83.85	83.88	84.69	88.84	81.85	**84.06**	84.18	8
IOB1-N	81.67	80.50	82.64	86.93	79.43	79.64	81.04	17.2
IOB2-F	85.11	82.92	85.56	87.95	80.27	81.49	82.92	14
IOB2-B	**86.07**	84.03	84.84	89.23	81.98	83.74	84.25	6.8
IOB2-N	81.60	83.53	84.44	88.72	81.11	82.65	83.55	13.4
IOE1-F	82.01	83.79	85.87	89.26	81.37	81.87	83.73	10.4
IOE1-B	82.64	83.80	84.65	88.69	81.74	83.77	84.04	10
IOE1-N	81.65	80.52	82.79	86.46	79.27	79.38	80.92	17.6
IOE2-F	82.70	84.65	85.90	89.94	82.68	81.78	84.36	4.8
IOE2-B	83.23	84.00	84.76	89.31	81.50	83.56	84.11	8.2
IOE2-N	81.60	84.11	**87.25**	89.35	82.77	82.09	84.30	4.8
SE-F	85.15	83.95	84.46	89.66	82.15	80.87	83.62	10
SE-B	85.90	83.67	85.52	89.46	81.85	83.11	84.03	8.4
SE-N	85.85	**85.49**	86.13	**90.47**	**83.27**	83.80	**85.83**	1.6

TABLE VI
EXPERIMENTAL RESULTS OF OUR PROPOSED METHODS AND SE-N OF THE BASE LINE ($F_{\beta=1}$): AV.($F_{\beta=1}$) AND AV.RANK ARE FOR FIVE PIECES DATA.

	CRL Cross	formal-run GENERAL	formal-run ARREST	domain-specific training	dry-run	dry-run training	Av. $F_{\beta=1}$	Av. rank
base line	85.85	85.49	86.13	90.47	83.27	83.80	85.83	5
Method 1	87.54(+1.69)	**87.20**	90.31	92.05	84.37	85.68	87.92 (+2.09)	2.8
Method 2	88.04(+2.19)	86.06	89.84	92.51	84.95	85.47	87.77 (+1.94)	3.4
Method 3	88.26(+2.41)	86.41	**91.85**	**92.61**	**85.49**	85.94	**88.46** (+2.63)	**1.6**
Method 4	**88.50**(+2.65)	87.09	90.20	91.78	**85.49**	**86.13**	88.14 (+2.31)	2

The goal is to find a decision function that predicts y for an unseen $x \in R^n$. A SVMs classifier gives the decision function $f(x) = sign(g(x))$ for an input vector x where

$$g(x) = \sum_{z_i \in SV} \alpha_i y_i K(x, z_i) + b.$$

$f(x) = +1$ means that x is a positive member, and $f(x) = -1$ means that is a negative member. The vectors z_i are called support vectors. $y_i \in \{-1, +1\}$ is the label of z_i and $0 < \alpha_i$ is the weight of z_i. Support vectors and other constants are determined by solving a quadratic programming problem. The $K(x, z)$ is a kernel function that maps vectors into a higher dimensional space. We use the polynomial kernel of degree 2 given by $K(x, z) = (1+x\cdot z)^2$ because NE extraction results with the polynomial kernel of degree 2 have shown the best performance in [6], [8], [9]. We used the soft-margin parameter with a value of 1, which is the same value used in past experiments with SVMs [6], [9].

We convert each set of features described in section II-B and II-C into a binary vector as the input of SVMs. We associate each feature with one of the elements of a vector space. Thus, the dimension of the vector space is equal to the number of the types of features. When converting a set of features into a binary vector, we set each element in a vector to 1 if the corresponding feature of the element is included in the set of features, and we set the other elements to 0.

We used the "one-versus-rest method" for extending binary classifiers to N-class classifiers, and prepared N binary classifiers, between a class and the remaining the classes.

This method may involve two or more classifiers which give $+1$ or no classifier give $+1$. In order to solve the problems, we used the Viterbi search. Since SVMs outputs are not probabilities, we use the sigmoid function $s(x) = 1/(1 + exp(-\beta x))$ with $\beta = 1$ to map $g(x)$ to a probability-like value [23].

TABLE VII
PERFORMANCE ON THE EIGHT IREX NE CLASS OF EACH EVALUATION DATA: THE TOP TABLE SHOWS THE RESULTS OF THE BASE LINE. THE BOTTOM TABLE SHOWS THE RESULTS OBTAINED WITH METHOD 3. BETTER PRECISION, RECALL AND F-MEASURE SCORES THAN BASE LINE ONES ARE IN BOLDFACE.

Base line										
	formal-run general		formal-run arrest		domain-specific training		dry-run		dry-run training	
	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.
ORGANIZATION	73.78	84.41	66.22	90.74	77.50	95.38	70.09	85.23	71.11	85.71
PERSON	88.45	89.46	85.57	89.25	97.87	94.85	92.31	92.86	90.58	88.03
LOCATION	80.77	87.96	86.79	87.62	87.27	87.80	86.98	88.83	89.02	86.97
ARTIFACT	36.73	48.65	53.85	43.75	45.45	62.50	9.52	25.00	28.36	70.37
DATE	93.86	95.94	95.83	93.24	98.55	97.14	82.73	85.85	90.51	96.88
TIME	94.92	98.25	94.74	100.00	94.44	100.00	87.50	91.30	87.50	100.00
MONEY	100.00	100.00	100.00	100.00	94.74	94.74	96.97	96.97	81.25	100.00
PERCENT	90.48	100.00	-	-	66.67	66.67	100.00	100.00	89.47	94.44
Total	82.54	88.65	83.80	88.59	88.89	92.10	79.37	87.57	79.59	88.48
F-measure	85.49		86.13		90.47		83.27		83.80	
Method 3										
ORGANIZATION	**76.61**	81.64	**83.78**	88.57	77.50	93.94	**74.77**	**86.49**	**72.59**	85.59
PERSON	**89.86**	**90.88**	**92.78**	**90.91**	**98.94**	**95.88**	**95.27**	**93.60**	**92.75**	87.67
LOCATION	**84.62**	**88.00**	**90.57**	**96.97**	**91.52**	**91.52**	**90.10**	88.27	**92.16**	**91.44**
ARTIFACT	**38.78**	46.34	**61.54**	**53.33**	**63.64**	**63.64**	**19.05**	**38.10**	**43.28**	69.05
DATE	**94.95**	95.64	**100.00**	**97.30**	**100.00**	**100.00**	**88.18**	85.09	**93.43**	**96.97**
TIME	**96.61**	**98.28**	**100.00**	**100.00**	94.44	**100.00**	**95.83**	85.19	**100.00**	88.89
MONEY	100.00	100.00	100.00	100.00	94.74	94.74	96.97	96.97	81.25	100.00
PERCENT	90.48	95.00	-	-	**100.00**	**75.00**	100.00	100.00	89.47	94.44
Total	**84.88**	88.00	**91.26**	**92.45**	**91.50**	**93.75**	**83.54**	87.53	**82.83**	**89.29**
F-measure	**86.41**(+0.92)		**91.85**(+5.72)		**92.61**(+2.14)		**85.49**(+2.22)		**85.94**(+2.14)	

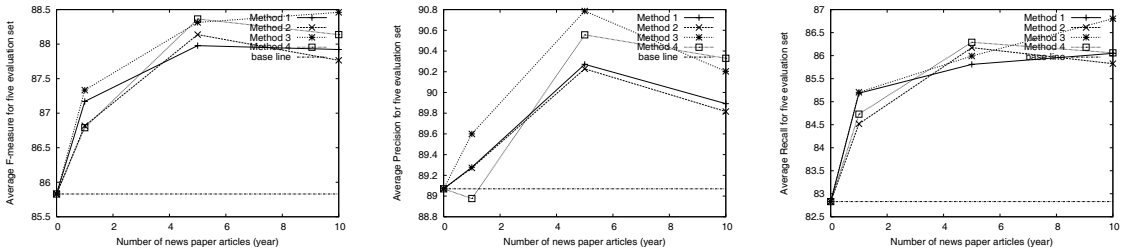

Fig. 3. Average F-measure, Precision and Recall obtained with NE extractors using NE-related labels of words collected from 1, 5 and 10 year news articles

III. FEATURE AUGMENTATION WITH UNLABELED DATA

In this section we describe NE-related labels collected from unlabeled data and the collection methods of the NE-related labels.

A. NE-related labels of words

We use NE-related information of words, which we call NE-related labels, as features. We collect the following information as NE-related labels from unlabeled data parsed with NE extractors.

- **Candidate NE class labels of words**: We collect the 33 types of NE labels defined by the SE representation as candidate NE labels of words. We also collect NE class labels of co-occurring words of each word by the same representation. We collect candidate NE labels of co-occurring words within two preceding and two succeeding words. We collect the information from the parsed results of unlabeled data with NE extractors.
  For example, as the candidate NE class labels of "田中*(Tanaka)*", B-ORG and S-PERSON are collected from the examples listed in Table III; as the candidate NE class labels of preceding words of "田中*(Tanaka)*", E-ORG and O for words appearing the next of "田中*(Tanaka)*", and O for the word two ahead position of "田中*(Tanaka)*" are collected.
- **Frequencies of candidate NE class labels**: These are the frequencies of the NE candidate labels of each word, which are counted from the parsed results. For example, as the frequencies for the candidate NE class labels of "田中*(Tanaka)*", "B-ORG" is counted once, "S-PERSON" is counted twice from the examples listed in Table III. For the candidate NE class labels of surrounding words of "田中*(Tanaka)*", "E-ORG" is collected once and "O" is collected twice as the candidate NE class labels of the next words position, and "O" is collected once as the candidate NE class labels of two ahead position. To express the frequencies of NE-related labels as binary features, we categorize the frequencies of these NE-related labels by n of each frequency; $n \leq 10$, $10 < n \leq 100$, and $100 < n$.

– **Ranking of NE-related labels**: This information is the ranking of candidate NE class labels for each word. Each ranking is decided with their frequencies counted from the parsed results. For the ranking in the candidate NE class labels of "田中*(Tanaka)*", "S-PERSON" is ranked as the first, and "B-ORG" is ranked as the second. As the ranking of the candidate NE class labels of words appeared in the next of "田中*(Tanaka)*", "O" is ranked as the first, and "E-ORG" is ranked as the second.

In this experimental setting, up to $495\ (= 3{\times}33{\times}5)$ features are collected for each word because we collect words with the candidate NE class labels of the two preceding and two succeeding word positions in addition to the current word ones.

We used the NE-related labels in the word-unit chunking as features. As a result, up to $2475\ (= 495 \times 5)$ features are added when training and extraction because we used features of words within 5-word window in addition to the current word features.

B. Collection methods of NE-related labels by using several NE extractors

We collect NE-related labels of words as follows:

– Step 1: Create NE extractors from a given training data by using SVMs. We create 15 NE extractors as described in Section II-A.
– Step 2: Extract NEs from unlabeled data with the NE extractors created in Step 1.
– Step 3: Collect NE-related labels from the automatically labeled data in Step 2.
– Step 4: Add the collected NE-related labels in Step 3 to the same training data used in Step 1 as new features.
– Step 5: Construct a new NE extractor from the training corpus created in Step 4 by using SVMs. As for the model of the new NE extractor, we use the same model of the best NE extractor model in the 15 extractor created at Step 1. When the new NE extractor extracts NEs, the collected NE-related labels are also used as features.

Some problems of using parsed results for collecting NE-related labels of words are noise and low coverage caused by wrong extraction. To avoid these problems as much as possible, we consider the following methods used at Step 3 and 4.

– **Method 1**: Collect NE-related labels of words from a parsed result with using an NE extractor. We use the NE extractor which shows the best accuracy of all 15 NE extractors and we use the F-measure ($F_{\beta=1}$) as a measure of accuracy. We think that the collected NE-related labels of words by this method archive moderately good for recall and precision.
– **Method 2**: Collect NE-related labels of words from each word chunk which all the NE extractors extract the word chunk as the same NE class. Even if one of the outputs of NE extractors to a word chunk is different from the outputs of the others, the word chunk is just ignored. By using this method, the collected NE-related labels of words have better precision than Method 1, but the recall is worse.
– **Method 3**: Collect NE-related labels from all the NE extractor outputs. By using this method, the collected NE-related labels of words are better recall than Methods 1 and 2, but the precision is worse than those two particular methods. When we use this method, we use the average frequencies of extracted results obtained from 15 NE extractor outputs as the frequencies of NE-related labels.
– **Method 4**: Collect NE-related labels of words by using Methods 1, 2 and 3, and all the NE-related labels are used as features differently. By using all the NE-related labels from Methods 1, 2 and 3, an NE extractor uses all their characteristics.

For collecting NE-related labels by using NE extractors based on distinct chunk representations as the SE representation, we convert the NE extractor outputs into the SE representation.

A problem of parsing large data by SVMs based NE extractors is the processing speed [6], [21]. To improve the processing speed of SVMs based NE extractors, we use Polynomial Kernel Expanded (PKE) method [21], which converts a kernel-based classifier into a simple linear classifier by expanding all feature combinations.

IV. EXPERIMENTS

A. Experimental Settings

We used the following data.

– Training and Evaluations data: We used the six pieces of Japanese NE data prepared by IREX [17]. We used CRL data for training and evaluation with cross-validation. We used the five data for evaluation, consisting of formal-run GENERAL, formal-run ARREST, domain-specific training data, dry-run data and dry-run training data. Table IV lists the statistics of NE types for each data set.
– Unlabeled data: We used 10-year period news articles: The news articles are the Mainichi Shinbun over a 10-year period between 1991 and 1993, 1995 and 1998, and 2000 to 2002. To keep the five pieces of evaluation data fully unseen in the training phases, we excluded the 1994 and 1999 news articles because they include the evaluation data.

We did the following experiments.

– Five-fold cross-validation on CRL data: We split CRL data into five pieces data. Each piece data consists of about $1/5$ articles included in CRL data. We separately collected NE-related labels for each classifier created from $4/5$ size of CRL data. We totally parsed $(75{*}10) = 750$ years amount of newspaper articles because we created $(15 * 5) = 75$ extractors for cross-validation. We

TABLE VIII
COMPARISON WITH RELATED WORKS: GE AND AR INDICATE GENERAL AND ARREST.

Method	GE	AR	CRL-DATA	NE extraction algorithm	Lexical resources
Uchimoto et al.[3]	80.17	85.75	-	Chunking by word with ME and transformation rules for word unit problems	Handcrafted NE dictionaries
Takemoto et al. [1]	83.86	-	-	Handcrafted Rules and Compound Lexicon for word unit problems	-
Utsuro et al. [24]	84.07	-	-	Combining three ME based NE extractor outputs by decision list based stacking	-
Yamada et al.[4]	-	-	83.2	Chunking by word with SVMs and examples in training data are segmented	-
Isozaki and Kazawa [6]	85.77	-	86.77	Chunking by word with SVMs and template rules for word unit problems	NTT Goi Taikei
Asahara and Matsumoto [8]	-	-	87.21	Chunking by character with SVMs using n-best results of morphological analysis	NTT Goi Taikei
Nakano and Hirai [9]	-	-	89.03	Chunking by character with SVMs using Japanese base phrase information	NTT Goi Taikei
Sasano and Kurohashi [25]	87.72	-	89.40	Chinking by word with SVMs using structural information	NTT Goi Taikei
Kazama and Torisawa [26]	-	-	88.93	Chunking by character with CRFs	Web documents and Wikipedia
our base line	85.49	86.13	85.85	Character-based chunking using outputs of a word-based NE extractor by stacking with SVMs	-
Method 1	**87.20**	**90.31**	87.54		unlabeled data
Method 2	**86.06**	**89.84**	88.04		unlabeled data
Method 3	**86.41**	**91.85**	88.26		unlabeled data
Method 4	**87.09**	**90.20**	88.50		unlabeled data

created 5 NE extractors based on character-unit chunking for each cross-validation.

– Evaluation on five pieces data: We created NE extractors by using CRL data, and we applied the NE extractors to the five pieces data.

We compared performance of NE extractors by using F-measure, which is defined as follows.
Recall = NUM / (the number of correct NEs),
Precision = NUM / (the number of NEs extracted by an NE extractor),
F-measure = 2 × Recall × Precision / (Recall + Precision), where NUM is the number of NEs correctly identified by an NE extractor.

B. Experimental Results

Table V lists the experimental results obtained with the 15 NE extractors. These NE extractors did not use NE-related labels presented in Section III-A as features. Of all 15 NE extractors, the SE-N based extractor, which is based on the SE representation without using NE labels assigned to preceding words as features, showed the best performance in the evaluation one the five pieces of data and the third best performance in the evaluation on cross-validation of CRL data. We used the SE-N based extractor results as the base line and the NE extractor for the Method 1. We also used SE-N model for implementing new NE extractors.

Table VI lists the experimental results obtained with NE extractors using NE-related labels of words collected from 10 year news articles by Methods 1 to 4 and the results of our base line.

All our methods exceeded the base line. These results show that NE-related labels of words collected from unlabeled data contributed to improved accuracy. Methods 2 to 4, which used NE-related labels collected with several extractors, showed better performance on cross-validation of CRL data than Method 1. Methods 3 and 4 showed better performance on five pieces data than Method 1. These results show that NE-related labels collected with several NE extractors contributed to improved accuracy.

Method 4 showed the best performance on five-fold cross validation, and the average F-measure was 2.65 points higher than the base line. Method 3 showed the best performance on five pieces data, and the average F-measure was 2.63 points higher than the base line.

Table VII lists that performance on eight NEs of each evaluation data. The results show that recalls of eight NEs are improved by using the NE-related labels while keeping higher precision in many cases.

Figure 3 shows the average F-measure, precision and recall for the five evaluation data with different amount of newspaper articles for collecting NE-related labels. Methods 1 to 4 based NE extractors with 1, 5 and 10 year news articles showed higher accuracy than the base line one. Furthermore, these results show that the higher recall than the base line while keeping higher precision except for Method 4 with 1 year news articles.

However, the precision scores of Methods 1 to 4 with 10 year news articles were worse than the precisions of Methods

1 to 4 with 5 year news articles. The F-measure scores of Methods 1, 2 and 3 were also worse than the corresponding Methods with 5 year news articles. A reason seems to be noise given by much number of NE-related labels of words.

One of the methods to solve the problem what we think is utilization of richer information as features, such as larger context information, phrase information [9] and dependency information, and so on, may be effective. The other is pruning of NE-related labels by using a threshold value like their frequencies and their ranking.

C. Comparison with Previous Work

Table VIII lists the results of the previous works using IREX Japanese NE extraction tasks. Compared to previous works, our base line showed relatively higher performance without using additional lexical resources. The result showed that the combination of a word-unit NE extraction and a character-unit NE-extraction is effective for Japanese Named Entity Extraction.

All the results obtained with our Methods 1 to 4, which use CRL data and unlabeled data of 10 year news articles for training, showed higher performance than the previous best results on IREX GENERAL and ARREST tasks. Uchimoto et al. [3] used a handcrafted NE dictionary and training data consisting of CRL data, dry-run data, dry-run training data, and domain-specific training data, for creating a maximum entropy (ME) model based NE extractor. Isozaki and Kazawa [6] used NTT GOI Taikei [27], which is a handcrafted thesaurus and CRL data for training. These results showed that NE-related labels of words collected from unlabeled data are useful just as well as handcrafted resources.

Our results showed higher performance than the handcrafted-rule based NE extractor [1] and the NE extraction combining outputs of three NE extractors by stacking [24].

Our methods showed better performance than a character-unit chunking using n-best morphological analysis results and NTT GOI Taikei as features [8]. However, our methods showed slightly worse performance than methods using Japanese base phrase information and NTT GOI Taikei [9], [25] and gazetteers induced from web documents and Wikipedia [26]. The reason seems to be the difference of features. We think that we can improve performance of our NE extractors by using features used in their character-unit chunking algorithms.

V. Conclusion

This paper proposes feature augmentation methods using unlabeled data and several Named Entity (NE) extractors. Our feature augmentation techniques collect candidate NE class labels of words and NE class labels of co-occurring words from unlabeled data parsed with several NE extractors. The experimental results with IREX GENERAL and ARREST tasks showed that NE extractors with our proposal methods showed higher performance than the previous best results using handcrafted lexical resources.

References

[1] Y. Takemoto, T. Fukushima, and H. Yamada, "A Japanese named entity extraction system based on building a large-scale and high quality dictionary and pattern-matching rules (in Japanese)," in *IPSJ Journal, 42(6)*, 2001, pp. 1580–1591.

[2] M. Collins and Y. Singer, "Unsupervised models for named entity classification," in *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999. [Online]. Available: citeseer.ist.psu.edu/collins99unsupervised.html

[3] K. Uchimoto, Q. Ma, M. Murata, H. Ozaku, M. Utiyama, and H. Isahara, "Named entity extraction based on a maximum entropy model and transformati on rules." in *Proc. of the ACL 2000*, 2000, pp. 326–335.

[4] H. Yamada, T. Kudoh, and Y. Matsumoto, "Japanese named entity extraction using Support Vector Machine (in Japanese)," in *IPSJ Journal, 43(1)*, 2002, pp. 44–53.

[5] X. Carreras, L. Màrques, and L. Padró, "Named entity extraction using adaboost," in *Proc. of CoNLL-2002*. Taipei, Taiwan, 2002, pp. 167–170.

[6] H. Isozaki and H. Kazawa, "Speeding up named entity recognition based on Support Vector Machines (in Japanese)," in *IPSJ SIG notes NL-149-1*, 2002, pp. 1–8.

[7] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang, "Named entity recognition through classifier combination," in *Proc. of CoNLL-2003*, 2003, pp. 168–171.

[8] M. Asahara and Y. Matsumoto, "Japanese named entity extraction with redundant morphological analysis," in *Proc. of HLT-NAACL 2003*, 2003, pp. 8–15.

[9] K. Nakano and Y. Hirai, "Japanese named entity extraction with bunsetsu features (in Japanese)," in *IPSJ Journal, 45(3)*, 2004, pp. 934–941.

[10] S. Miller, J. Guinness, and A. Zamanian, "Name tagging with word clusters and discriminative training." in *HLT-NAACL*, 2004, pp. 337–342.

[11] D. Freitag, "Trained named entity recognition using distributional clusters," in *Proc. of EMNLP 2004*. Association for Computational Linguistics, July 2004, pp. 262–269.

[12] R. Ando and T. Zhang, "A high-performance semi-supervised learning method for text chunking," in *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 1–9. [Online]. Available: http://www.aclweb.org/anthology/P/P05/P05-1001

[13] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. of ACL-1995*, 1995, pp. 189–196.

[14] E. Riloff and R. Jones, "Learning dictionaries for information extraction by multi-level bootstrapping," in *AAAI/IAAI*, 1999, pp. 474–479. [Online]. Available: citeseer.ist.psu.edu/article/riloff99learning.html

[15] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. of the 11th COLT*, 1998, pp. 92–100.

[16] R. K. Ando, "Semantic lexicon construction: Learning from unlabeled data via spectral analysis," in *Proc. of CoNLL-2004*. Boston, MA, USA, 2004, pp. 9–16.

[17] C. IREX, *Proc. of the IREX workshop*, 1999.

[18] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Proc. of the Third Workshop on Very Large Corpora*. Association for Computational Linguistics, 1995, pp. 82–94. [Online]. Available: citeseer.ist.psu.edu/article/ramshaw95text.html

[19] E. Tjong Kim Sang and J. Veenstra, "Representing text chunks." in *Proc. of EACL '99*, Bergen, Norway, 1999. [Online]. Available: http://www.cnts.ua.ac.be/Publications/1999/TV99

[20] T. Kudo and Y. Matsumoto, "Chunking with Support Vector Machines," in *Proc. of NAACL 2001*, 2001.

[21] ——, "Fast methods for kernel-based text analysis," in *Proc. of ACL-2003*, 2003, pp. 24–31.

[22] V. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, 1998.

[23] J. C. Platt, *Probabilities for SV machines*, A. J. Smola, P. L. Bartlett, B. Sch¨olkopf, and D. Schuurmans, Eds. MIT Press, 2000.

[24] T. Utsuro, M. Sassano, and K. Uchimoto, "Combining outputs of multiple Japanese named entity chunkers by stacking," in *Proc. of EMNLP 2002*, 2002, pp. 281–288.

[25] R. Sasano and S. Kurohashi, “Japanese named entity recognition using structural natural language processing,” in *Proc. of IJCNLP’08*, 2008, pp. 607–612.
[26] J. Kazama and K. Torisawa, “Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations,” in *Proc. of ACL-08: HLT*, 2008, pp. 407–415.
[27] S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi, *Goi-Taikei -A Japanese Lexicon CDROM*. Iwanami Shoten, 1999.