

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 learning 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 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-Hiragana². 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.

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

²The 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+”.

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

NE / Data	Training	Evaluation					Total
	CRL data	formal-run GENERAL	formal-run ARREST	domain-specific training	dry-run	dry-run training	
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 $f(x) = \text{sign}(g(x))$ 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].

- **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

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 *IPSI 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 transformation 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 *IPSI 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 *IPSI 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 *IPSI 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. Schölkopf, 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.

