Named Entity Recognizer Trainable from Partially Annotated Data

Tetsuro Sasada*, Shinsuke Mori*, Tatsuya Kawahara* and Yoko Yamakata†
*Academic Center for Computing and Media Studies, Kyoto University,
Yoshidahonmachi, Sakyo-ku, Kyoto, 606-8501, Japan
Email: {sasada@ar.media.kyoto-u.ac.jp, forest@i.kyoto-u.ac.jp, kawahara@i.kyoto-u.ac.jp}
†Graduate School of Informatics, Kyoto University,
Yoshidahonmachi, Sakyo-ku, Kyoto, 606-8501, Japan
Email: yamakata@dl.kuis.kyoto-u.ac.jp

***Abstract*—In this paper we propose a named entity recognizer (NER) which we can train from partially annotated data. As the natural language processing is getting to be applied to diverse texts, there arise high demands for the NER for new named entity (NE) definition in different domains. For these special NE definitions, only a small annotated corpus is available in the beginning, and a rapid and low-cost development of an NER is needed in practice. To satisfy the needs, we propose the use of partially annotated data, which is a set of sentences in which only a limited number of words are annotated with NE tags. Our NER method uses two-pass search for sequential labeling of NE tags: (1) enumerate NE tags with confidences for each word independently from the tags for other words and (2) the best NE tag sequence search referring to the tag-confidence pairs by CRFs. For the first-pass module, our method uses partially annotated data to improve the accuracy in the target domain. By this two-pass search framework, our method is expected to incorporate tag sequence statistics and to outperform state-of-the-art NERs based on a sequence labeling while keeping the high domain adaptability. We conducted several experiments comparing state-of-the-art NERs in various scenarios. The results showed that our method is effective both in the normal case and in adaptation cases.**

***Keywords*-Partial annotation; Incomplete data; Named entity recognition; Pointwise prediction; Sequence labeling; Recipe**

I. Introduction

One of the important natural language processing (NLP) is to recognize spans of words in a text corresponding to the real world and classify them into one of the classes defined in advance. In newspaper articles, which were the main target of NLP for long time, the classes are person names, company names, amount of money, etc. as defined in [1]. In the researches they are called named entities (NEs) and the task of automatically recognizing them is called the NE recognition (NER). NERs are useful for information retrieval from newspaper articles, question and answering about the world knowledge, and others. The NER task can be considered as a sequence labeling and tried many methods such as hidden markov model, conditional markov model, support vector machine (SVM) with dynamic programming (DP), conditional random fields (CRFs), etc. [2] [3] [4].

As the NLP is getting to be used more and more widely, the NER is applied to various texts in many languages. In addition, NLP users started to notice that task dependent definitions of NEs are useful for a special purpose instead of the general definition. Famous one is medical NE [5]. Obviously body part names, disease names, and protein names are important for information retrieval or text mining in medical texts. Nowadays there are many applications of NLP. For the reputation analysis of a company it is important to distinguish the product names of the company from those of its competitors. For recipe search it is important to recognize food name correctly in a certain context. For example, a recipe entitled "hamburger of steak house" is not a steak recipe. So we want to figure out that "steak" in this context is not a food. For these special NE definitions in the beginning only a small annotated corpus is available and a rapid and low cost development of an NER is called for.

In this background we propose an NER which we can train from partially annotated data. In NER case partially annotated data is a set of sentences in which only some words are annotated with NE tags and others are not. In practical cases they may be new NEs not appearing in a small fully annotated data. Our method is composed of two modules: (1) Enumerate NE tag with confidence for each word independently from the tags for other words and (2) NE tag sequence search referring to the tag-confidence pairs. We conducted several experiments comparing state-of-the-art NERs in various scenarios. The results showed that our method is effective both in the normal case and in adaptation cases.

II. Related work

The task we solve in this paper is NER. NER is a sequence labeling problem and many solutions have been proposed [2], [6, *inter alia*]. To our best knowledge one of the state-of-the-art methods is based on CRFs [7]. In this method first they convert the training corpus annotated with NE tags into so-called extended BIO system. B, I, and O stand for begin, intermediate, and other, respectively. Let us assume that there are NE types T_1, T_2, $\ldots$, T_J, they annotate a word sequence w_1, w_2, $\ldots$, w_n of an NE of type T_j as w_1/T_j-B, w_2/T_j-I, $\cdots$, w_n/T_j-I and a word not included in any NE as $w/$O. In the BIO system, there are $2J+1$ tags and

a word is annotated with one of them. The problem is similar to POS tagging which assign a grammatical category tag to each word, but in NER there are constraints on tag sequence. For example, (w_i/O, w_{i+1}/T$_j$-I), (w_i/T$_j$-B, w_{i+1}/T$_k$-I) and (w_i/T$_j$-I, w_{i+1}/T$_k$-I) ($j \neq k$), are illegal and we cannot interpret them. The NER based on CRFs automatically captures these constraints and outputs a legal sequence for an input word sequence. Some researches use a pointwise (PW) classifier such as a SVM or a logistic regression (LR) combined with a tag sequence search module based on dynamic programing (DP) .

These NERs based on SVM+DP or LR+DP has an advantage that they can use a partially annotate data for training, in which only some words are annotated with BIO tags and many other words are not. This advantage is very beneficial especially in resource-poor situations such as NER for a new NE definition or a new language. As it is well known, the coverage has a strong relationship with the NER accuracy. And the trainability from partial annotations allows annotators to focus on new NEs or an active learning [8] [9] [10] [11] [12] to select annotation unit smaller than a sentence. These methods help us to build an NER for shorter time and lower cost. This advantage may also allow researchers to try to devise an NER method for using natural annotations like HTML tags in Wikipedia, which is a hot topic in the word segmentation research recently [13] [14]. Our method extends these pointwise NERs with a reranker based on CRFs. Our BIO tag sequence search is more accurate than DP, so our method is expected to be better than the pointwise NERs without losing their advantage, trainability from partially annotate data.

As the input of our NER we assume a word sequence but not tagged with a part-of-speech (POS) tag. So when we apply our method to languages without obvious word boundary we need a word segmenter [15] [16]. The reason why we do not assume POS tagger results is that some research has reported a severe degradation in the POS tagger accuracy on texts in a new domain [17]. By assuming a word sequence as the input, we can skip an adaptation of a POS tagger to the target domain. As a result we can avoid that an entire NLP system including our NER loses its domain adaptability in real use.

The domain in which we test our NER in the experiment is cooking recipe. In the past the main target of NLP was newspaper articles but nowadays NLP is used in various texts. For example, a special definition of NE for medical texts has been defined and medical NER had a great success [18] [5]. Our application, recipe texts, is one of the user generated contents and have many potential applications ranging from researches to real uses: recipe search [19], recipe summarization [20], cooking help system [21], procedural text understanding [22] [23], computer vision [24], [25], cooking robot [26], etc. The recipe NER has not been as mature as the medical NER and only small training data is available. So it is a good test bet for an NER trainable from various types of training data, which is important in resource-poor domain and/or language. The NER method which we propose in this paper is not limited to this domain but is applicable to others.

Table I
R-NE TAGS.

r-NE tag	Meaning
F	Food
T	Tool
D	Duration
Q	Quantity
Ac	Action by the chef
Af	Action by foods
Sf	State of foods
St	State of tools

III. RECIPE NAMED ENTITY

The test data we use in the experiment is named entity specially defined for recipe texts (r-NE) [28]. Their structure is the same as the general NE [1]. An r-NE is a span of one or more words without overlap. No NE boundary occurs in the middle of a word. So a word in a sentence belongs to at most one r-NE. An r-NE has one type label listed in Table I. So we can say that only the type definition is different from the general NE. The types for the general NE are designed to be useful for information retrieval from the newspaper articles. Contrary r-NE types are useful to recognize actions, objects, and their status in the recipe texts. They are important for recipe text search [19], its understanding [26], and symbol grounding for cooking videos [24], [25]. Similar to NER for the general NE, NER for r-NE can be formalized as a sequence labeling problem and many solutions for the general NE [2], [6, *inter alia*] are applicable.

The reason why we use r-NE instead of the general NE is that we want to solve a practical problem, in which some NLP application is under development and we want to increase the NER accuracy in a resource-poor situation. Our NER method is, however, applicable to the general NE and other NE such as medical NE etc.

IV. 2-STEP NAMED ENTITY RECOGNITION

The NER method which we propose in this paper is composed of two modules:

1) Enumerate BIO tag with confidence for each word independently from the tags for other words,
2) BIO tag sequence search referring to the tag-confidence pairs.

The procedure is similar to a POS tagger trainable from partially annotated sentences [27]. In NER, however, the second process is necessary to output a consistent tag seqquence. In this section we explain these one by one.

Table II
FEATURE SET OF THE LR.

Type	Feature templates
Character n-gram	x^{-1}, x^{+1}, $x^{-2}x^{-1}, x^{-1}x^{+1}, x^{+1}x^{+2}$ $x^{-2}x^{-1}x^{+1}, x^{-1}x^{+1}x^{+2}$
Character type n-gram	$c(x^{-1}), c(x^{+1})$, $c(x^{-2})c(x^{-1}), c(x^{-1})c(x^{+1}), c(x^{+1})c(x^{+2})$, $c(x^{-3})c(x^{-2})c(x^{-1}), c(x^{-2})c(x^{-1})c(x^{+1})$, $c(x^{-1})c(x^{+1})c(x^{+2}), c(x^{+1})c(x^{+2})c(x^{+3})$

Table III
FEATURE SET OF THE CRFS.

Type	Feature templates
Word n-gram	w^{-1}, w^{+1}, $w^{-2}w^{-1}, w^{-1}w^{+1}, w^{+1}w^{+2}$
Word type n-gram	$c(w^{-1}), c(w^{+1})$, $c(w^{-2})c(w^{-1}), c(w^{-1})c(w^{+1})$, $c(w^{+1})c(w^{+2})$, $c(w^{-2})c(w^{-1})c(w^{+1})$, $c(w^{-1})c(w^{+1})c(w^{+2})$
Tag-confidence pair (LR+CRF only)	$\langle t_1, s_{i,1}\rangle, \langle t_2, s_{i,2}\rangle, \cdots, \langle t_{2J+1}, s_{i,2J+1}\rangle$

A. Tag-confidence Pair Enumeration

Given an input word sequence, the first module provides pairs of a tag and its confidence for each word to the second module. In order to make this module trainable from partially annotated data, we propose to use a pointwise classifier which refers, as features, only to the information contained in the input word sequence but not to the estimation results (so-called dynamic features).

As it is clear from the above design, this module is trainable from partially annotated data, because we can just use only the annotated words and its context as the training data. The following example:

ex.) Sprinkle black/F-B pepper/F-I and salt,

where only two words are annotated with BIO tags, is converted into the training data as follows:

left context	word	right context	tag
$\langle BOS\rangle$	Sprinkle	black pepper	–
$\langle BOS\rangle$ Sprinkle	black	pepper and	F-B
Sprinkle black	pepper	and salt	F-I
black pepper	and	salt $\langle EOS\rangle$	–
pepper and	salt	$\langle EOS\rangle$	–

We train a pointwise classifier such as SVM or LR [29], which estimate the tag for a word.

At runtime, different from the normal classification task, the classifier enumerates all the possible tags and their confidence. As the confidence we can use the margin from the separation hyper-plain in the SVM case or probability in the LR case. In this paper we use an LR as the classifier and the confidence $s_{i,j}$ for each tag t_j for a word w_i in the context of $\boldsymbol{x}^-, w_i, \boldsymbol{x}^+$ is calculated as follows:

$$s_{i,j} = P_{LR}(t_j|\boldsymbol{x}^-, w_i, \boldsymbol{x}^+).$$

The features are listed in Table II. $c(\cdot)$ is a function, which maps the character type of a word or a character (Chinese character, *hiragana*, Arabic number, etc.). So we have $(\langle t_1, s_{i,1}\rangle, \langle t_2, s_{i,2}\rangle, \cdots, \langle t_{2J+1}, s_{i,2J+1}\rangle)$, where $2J+1$ is the size of the BIO tag set and $s_{i,j}$ is the confidence of BIO tag t_j for word w_i.

B. Search for the Best Sequence

The second module is to search the best tag sequence given a word sequence annotated with tag-confidence pairs provided by the first module.

	$P_{LR}(t\|w)$	w = Sprinkle	black	pepper	and	salt
	F-B	0.00	**0.40**	0.37	0.00	**0.80**
	F-I	0.00	0.10	**0.63**	0.00	0.20
	Ac-B	**0.99**	0.00	0.00	0.00	0.00
t	Ac-I	0.01	0.00	0.00	0.00	0.00
	T-B	0.00	0.50	0.00	0.00	0.00
	⋮	⋮	⋮	⋮	⋮	⋮
	O	0.00	0.00	0.00	**1.00**	0.00

Figure 1. DP search for the most likely tag sequence.

1) DP Search: We can use a DP search to select the most likely tag sequence, where the likelihood can be defined the product of the probabilities as follows:

$$\hat{\boldsymbol{t}}_1^m = \underset{t_1,\, t_2,\, \ldots,\, t_m}{\textbf{argmax}} \prod_{j=1}^{m} s_{i,j}.$$

Figure 1 illustrates this DP search. The numbers in the bold face indicate the selected node. In the search, we take the constraints on the BIO tag sequence into consideration. For example, as shown in Figure 1, "black/T-B pepper/F-I" is illegal. This is one of the baselines and we test this in the experiments.

2) Sequence Labeling: Instead of the naive DP search, we propose to use a sequence labeling to search for the best sequence. By using a sequence labeling based on machine learning techniques we can take more context information into consideration such as tag sequence tendency for a certain word sequence, etc.

The input of this module at runtime is a word sequence annotated with tag-confidence pairs provided by the first module. Since at runtime the word sequence in focus is new for the first module, we have to emulate this situation at the training time of the second module for the model to be effective for new texts. Thus we execute the following procedures:

(i) Divide the training corpus into N parts of equal size,
(ii) Build the i-th pointwise classifier from $N-1$ parts of training corpus excluding the i-th part, and
(iii) Enumerate all the BIO tags with their confidence for each word in the i-th part by using the i-th pointwise

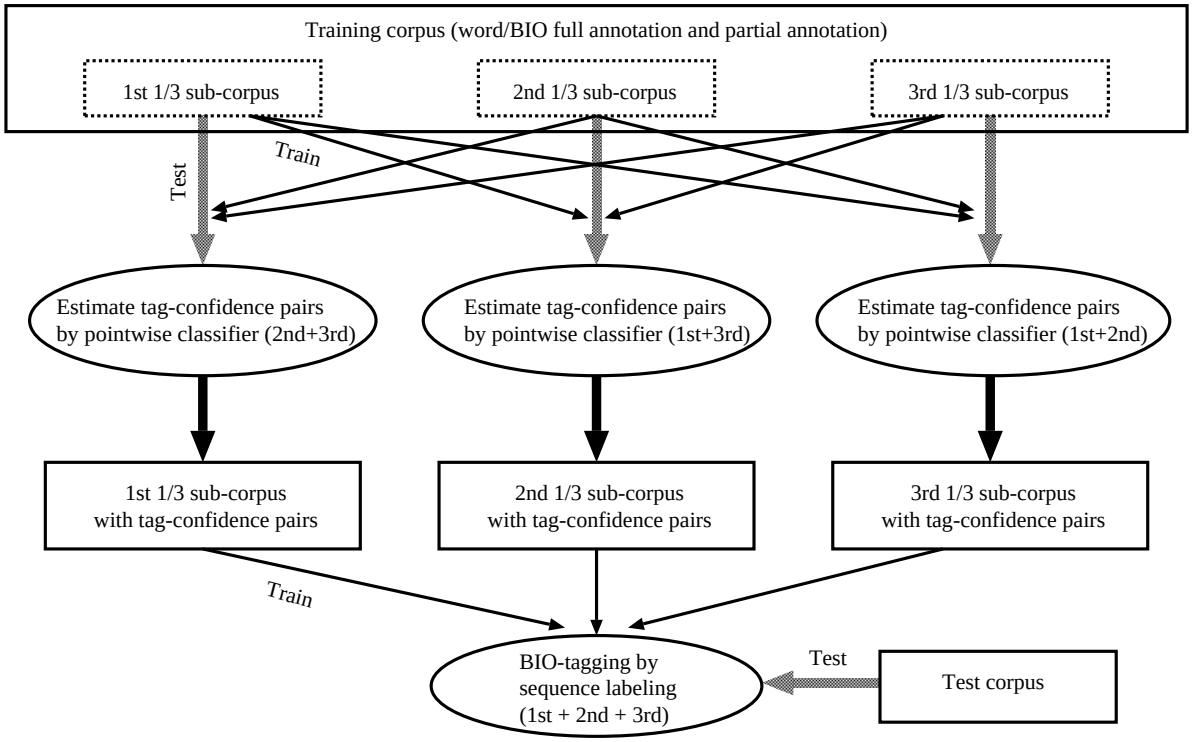

Figure 2. Procedure for generating the training corpora for BIO tagging by the sequence labeling ($N = 3$).

classifier.

Figure 2 shows these procedures. As a result we can annotate the words in the training corpus with tag-confidence pairs $(\langle t_1, s_{i,1}\rangle, \langle t_2, s_{i,2}\rangle, \cdots, \langle t_{2J+1}, s_{i,2J+1}\rangle)$ estimated by a pointwise classifier built from training data not containing the words in focus[1]. That is to say, we can successfully emulate the runtime situation.

Now we are ready to train the second module. This part is formulated as a sequence labeling. The training data is a set of word sequences annotated with tag-confidence pairs provided by the first module. The training data is similar to Figure 1 except that the correct tags for each word (label sequence) are attached in addition. As the sequence labeling we use CRFs [7], but we can use any other sequence labeling methods such as structured SVM [31]. Table III lists the features referred to by the second module. Note that in many NER researches POSs of the word in focus or in the context are also referred to as features, but we do not do it to keep the domain adaptability of our method in the entire system as we stated in Section II. If we assumed the POS as the input, we would have to spend time and cost to adapt a POS tagger to the target domain in order to build a practically valuable NER system.

[1]We can also use so-called leaving-one-out technique [30], but it is computationally too costly because we have to build as many models as the number of words in the training data.

Table IV
CORPUS SPECIFICATION.

Usage	#recipes	#sentences	#r-NEs	#words	#characters
Train	386	2,946	17,243	54,470	82,393
Test	50	371	1,996	6,072	9,167
(Total)	436	3,317	19,239	60,542	91,560

Table V
EXPERIMENTAL SETTINGS ABOUT TRAINING CORPUS.

training corpus set	#sentences	#r-NEs	#BIO tags
1/2 FULL	1,473	8,543	27,119
1/2 FULL + 1/2 PART	2,946	10,810	31,770
1/1 FULL	2,946	17,243	54,470

V. EVALUATION

As evaluations of our NER, we measured the accuracies of our NER and other methods under various settings. In this section we present the results and evaluate our NER.

A. Experimental Settings

The domain in the experiments below is cooking recipe. The NE definition is described in Section III which is different from the general one for newspaper articles [1]. So we test our method mainly in a relatively resource-poor situation. The corpus we used is procedural text sentences fully annotated with r-NE [28]. Table IV shows the

Table VI
RESULT:1/2 FULL ANNOTAION CORPUS.

method	BIO Accu.	Precision	Recall	F-measure
CRF	0.8949	0.8491	0.8372	0.8438
LR	0.8930	0.8441	0.8407	0.8424
LR+DP	0.8951	0.8397	0.8477	0.8437
LR+CRF (proposed)	**0.8989**	**0.8591**	0.8402	**0.8495**

Table VII
RESULT: 1/2 FULL AND 1/2 PARTIAL ANNOTAION CORPUS.

method	BIO Accu.	Precision	Recall	F-measure
CRF	0.8990	0.8612	0.8452	0.8531
LR	0.8995	0.8559	0.8452	0.8505
LR+DP	0.9012	0.8539	0.8552	0.8546
LR+CRF (proposed)	**0.9112**	**0.8773**	**0.8632**	**0.8702**

Table VIII
RESULT: 1/1 FULL ANNOTAION CORPUS.

method	BIO Accu.	Precision	Recall	F-measure
CRF	0.9065	0.8759	0.8627	0.8693
LR	0.9056	0.8713	0.8582	0.8647
LR+DP	0.9069	0.8696	0.8652	0.8674
LR+CRF (proposed)	**0.9157**	**0.8853**	**0.8742**	**0.8798**

specifications of the corpus. We see in this paper that the number of sentences, 2,946 + 371, is much smaller than the corpus annotated with general domain NE (normally more than 10,000 sentences). In addition the sentences tend to be much shorter than newspaper articles, thus the number of annotated NE instances is much smaller that the general NER case. For a detailed description about the NE definition and the corpus the readers may refer to [28].

In the experiments we divided the training data into two parts in order to test our NER and others simulating resource-poor situations, or the beginning of a project which NLP is applied to. The concrete settings are as follows:

- 1/2 FULL: The first half of training data is available as fully annotated corpus,
- 1/2 FULL + 1/2 PART: In addition to the first half, the second half is available as a partially annotated corpus,
- 1/1 FULL: The entire training data is available as fully annotated corpus, that is the training data size is twice as large as the 1/2 FULL case.

Table V shows the numbers of r-NEs and those of BIO tags in the above settings. In the partially annotated corpus, 1/2 PART, we emulated the situation where new r-NEs not contained in 1/2 FULL are annotated three times (if the frequency is less than 3, that number of times).

The methods we compared are as follows:

- **CRF**: Sequence labeling by conditional random fields trainable from partially annotated data [32],
- **LR**: Pointwise classification by a logistic regression [29] without DP search,
- **LR + DP**: LR with DP search,
- **LR + CRF**: LR with the best tag sequence search by conditional random fields trained from the fully annotate data only (proposed method; see Section IV).

In **LR + CRF**, we divided the fully annotate training into 3 to create the corpus containing sentences of words with tag-confidence pairs (see Section IV-B and Figure 2).

As the implementation of CRFs which we can train from partially annotated data [32] we used partial-crfsuite toolkit [2] [14]. As an LR classifier we adopt KyTea toolkit[3] [16]. Table III and II show the feature sets of **CRF** and **LR**, respectively, respectively. The 2nd module of **LR + CRF** uses the tag-confidence pairs as features in addition.

[2]https://github.com/ExpResults/partial-crfsuite

[3]http://www.phontron.com/kytea/

B. Evaluation Criterion

We adopt two criteria. The first one is the tag accuracy, the percentage of the BIO tags correctly estimated by the NER system. The second is F-measure, which is the standard criterion for the NER task. The F-measure is the harmonic mean of precision and recall. Let N_{sys}, N_{ref}, and N_{int} be the number of the estimated NEs, the gold standard NEs, and their intersection, respectively. Then precision = N_{int}/N_{sys}, recall = N_{int}/N_{ref}, and F-measure = $2N_{int}/(N_{ref}+N_{sys})$, the harmonic mean of them.

C. Evaluation

We compared our method **LR + CRF** with three methods: **CRF**, **LR**, and **LR + DP** under three settings, 1/2 FULL, 1/2 FULL + 1/2 PART, and 1/1 FULL. Table VI, VII, and VIII show the results. And Figure 3 shows the F-measures of the same results in graph form. As we see in Figure 3, the proposed method, **LR + CRF**, outperforms the other three methods, **CRF**, **LR**, and **LR + DP**) in all the cases. Below we discuss the results in detail.

When a large full annotation corpus is available, that is the case of Table VIII, **CRF** is better than **LR**, and **LR + DP**. This is the reason why CRF is used as the state-of-the-art method for NER task in recent researches [33]. However, in case where the size of the full annotation corpus is small (Table VI) or a partially annotated corpus is available additionally (Table VII), **LR + DP** is better than **CRF**. **LR + DP** is simple and not so bad because the machine learning part is pointwise, not sequence labeling, thus its training time is much shorter than **CRF** especially when a partially annotated corpus is available. As we see in the paper [32], training CRF from a partially annotated corpus requires a number of iterations calculating the expected values of the possible tags for each words without annotation and the time needed for training tends to be long. Contrary **LR** is based on a pointwise classifier and we can train it for short time just by using the annotated words [34]. Thus we can say that in the beginning of an NE tagging project with a new

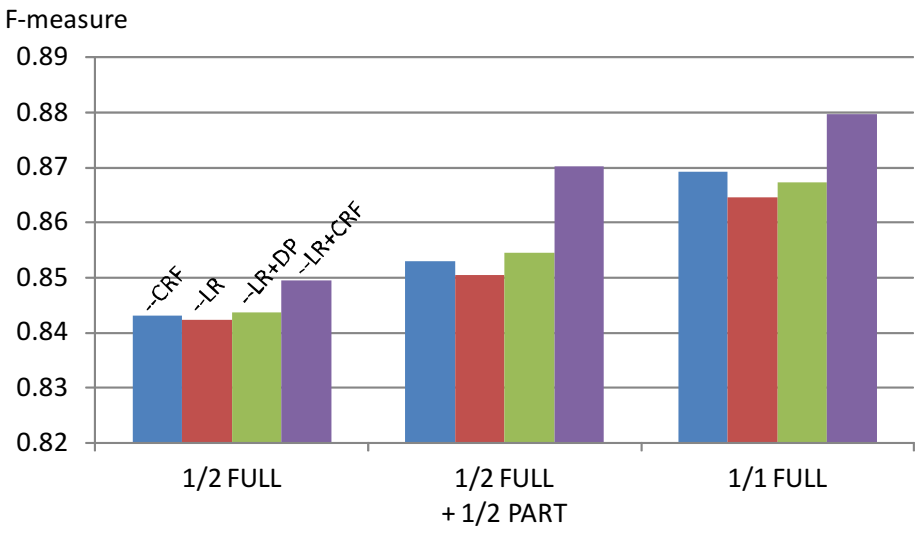

Figure 3. NER accuracies.

NE definition, **LR + DP** is suitable allowing frequent model updates especially when use an active learning technique [8] [16].

In real situations, we want to maximize the accuracy for a certain annotation cost. As we have pointed out, it is good to concentrate annotation work on informative words. One simple strategy is to annotate new r-NEs for a few times to increase the coverage. 1/2 FULL + 1/2 PART represents this situation, where new r-NEs are annotated at most for three times[4]. As we have pointed out above, our method, **LR + CRF**, is the best in this case as well. The important point is, however, the differences in F-measure between **LR + CRF** and the others are very large in this case (see Figure 3). This result indicates that our method is effective for constructing an NE recognizer in real situations. Surprisingly Figure 3 clearly shows that **LR + CRF** trained from 1/2 FULL + 1/2 PART is better than the others trained from 1/1 FULL. As shown in Table V, the number of additional r-NE annotations for 1/2 FULL + 1/2 PART from 1/2 FULL (2,267 = 10,810 - 8,543) is around a quarter of 1/1 FULL from 1/2 FULL (8,700 = 17,243 - 8,543). So we can say that with **LR + CRF** we need less annotation work to achieve a higher accuracy. In addition the time needed for training **LR + CRF** from a partially annotated corpus is as short as **LR** and **LR + DP**, and much shorter than **CRF**, because we only need to update the first part, the pointwise classifier which is the same as those in **LR** and **LR + DP**. Therefore we can say that after development of a small fully annotated corpus it is a good strategy to annotate new NEs providing a partially annotated corpus and to use our method, **LR + CRF**, which is trained from that partially annotated corpus.

[4]This is a simulation and does not include real annotation work. An experiment with the real annotation time is a future work.

VI. Conclusion

In this paper we have proposed a method for recognizing named entities. Our method is trainable from partially annotated data and we have experimentally shown that our method is better than extisting ones in both the situations where only fully annotated data is available and where partially annotated data is additionally available. Thus our method is useful not only for the normal setting but also for resource-poor domains and/or languages.

An interesting research direction is to try to improve NER by using partially annotated texts converted from wikipedia or other hyper texts. Active learning is another good research direction. Our method allows more flexible units to be annotated selection to make active learning more effective.

Acknowledgement

This work was supported by JSPS Grants-in-Aid for Scientific Research Grant, and JSPS Grant-in-Aid for Young Scientists Grant. We are grateful to the annotators for their contribution to the design of the guidelines and the annotation effort.

References

[1] N. A. Chinchor, "Overview of muc-7/met-2," in *Proceedings of the Seventh Message Understanding Conference*, 1998.

[2] A. Borthwick, "A maximum entropy approach to named entity recognition," Ph.D. dissertation, New York University, 1999.

[3] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, 2005, pp. 363–370.

[4] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the Seventh Conference on Computational Natural Language Learning*, 2003.

[5] A. Ben Abacha and P. Zweigenbaum, "Medical entity recognition: A comparaison of semantic and statistical methods," in *Proceedings of BioNLP 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 56–64. [Online]. Available: http://www.aclweb.org/anthology/W11-0207

[6] E. F. T. K. Sang and F. D. Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Computational Natural Language Learning*, 2003, pp. 142–147.

[7] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth ICML*, 2001.

[8] B. Settles, M. Craven, and L. Friedland, "Active learning with real annotation costs," in *NIPS Workshop on Cost-Sensitive Learning*, 2008.

[9] M. Sassano, "An empirical study of active learning with support vector machines for japanese word segmentation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 505–512.

[10] K. Tomanek and U. Hahn, "Semi-supervised active learning for sequence labeling," in *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, 2009, pp. 1039–1047.

[11] M. Tang, X. Luo, and S. Roukos, "Active learning for statistical natural language parsing," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 120–127.

[12] Y. S. Chan and H. T. Ng, "Domain adaptation with active learning for word sense disambiguation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007, pp. 49–56.

[13] F. Yang and P. Vozila, "Semi-supervised chinese word segmentation using partial-label learning with conditional random fields," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 90–98.

[14] Y. Liu, Y. Zhang, W. Che, T. Liu, and F. Wu, "Domain adaptation for crf-based chinese word segmentation using free annotations," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 864–874.

[15] R. Sproat and C. S. W. G. N. Chang, "A stochastic finite-state word-segmentation algorithm for chinese," *Computational Linguistics*, vol. 22, no. 3, pp. 377–404, 1996.

[16] G. Neubig, Y. Nakata, and S. Mori, "Pointwise prediction for robust, adaptable japanese morphological analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 2011, pp. 529–533.

[17] G. Kevin, S. Nathan, O. Brendan, D. Dipanjan, M. Daniel, E. Jacob, H. Michael, Y. Dani, F. Jeffrey, and S. N. A, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *Proceedings of the ARPA Workshop on Human Language Technology*, 2011, pp. 42–47.

[18] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 147–155. [Online]. Available: http://www.aclweb.org/anthology/W09-1119

[19] L. Wang, Q. Li, N. Li, G. Dong, and Y. Yang, "Substructure similarity measurement in chinese recipes," in *Proceedings of the 17th International Conference on World Wide Web*, 2008, pp. 978–988.

[20] Y. Yamakata, S. Imahori, Y. Sugiyama, S. Mori, and K. Tanaka, "Feature extraction and summarization of recipes using flow graph," in *Proceedings of the 5th International Conference on Social Informatics*, ser. LNCS 8238, 2013, pp. 241–254.

[21] A. Hashimoto, N. Mori, T. Funatomi, Y. Yamakata, K. Kakusho, and M. Minoh, "Smart kitchen: A user centric cooking support system," in *Proceedings of the 12th Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2008, pp. 848–854.

[22] Y. Momouchi, "Control structures for actions in procedural texts and pt-chart," in *Proceedings of the Eighth International Conference on Computational Linguistics*, 1980, pp. 108–114.

[23] H. Maeta, T. Sasada, and S. Mori, "A framework for recipe text interpretation," in *Proceedings of the SixthInternational Workshop on Cooking and Eating Activities*, 2014.

[24] I. Naim, Y. C. Song, Q. Liu, H. Kautz, J. Luo, and D. Gildea, "Unsupervised alignment of natural language instructions with video segments," in *Proceedings of the 28th National Conference on Artificial Intelligence*, 2014.

[25] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," in *Proceedings of the 14th International Conference on Computer Vision*, 2013.

[26] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus, "Interpreting and executing recipes with a cooking robot," in *Proceedings of The 13th International Symposium on Experimental Robotics*, 2013, pp. 481–495.

[27] S. Mori, Y. Nakata, G. Neubig, and T. Sasada, "Pointwise prediction and sequence-based reranking for adaptable part-of-speech tagging," in *Proceedings of the Eleventh International Conference Pacific Association for Computational Linguistics*, 2015.

[28] S. Mori, H. Maeta, Y. Yamakata, and T. Sasada, "Flow graph corpus from recipe texts," in *Proceedings of the Nineth International Conference on Language Resources and Evaluation*, 2014, pp. 2370–2377.

[29] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[30] R. Kneser and H. Ney, "Improved clustering techniques for class-based statistical language modelling," in *Proceedings of the Third European Conference on Speech Communication and Technology*, 1993, pp. 973–976.

[31] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Machine Learning*, vol. 6, pp. 1453–1484, 2005.

[32] Y. Tsuboi, H. Kashima, S. Mori, H. Oda, and Y. Matsumoto, "Training conditional random fields using incomplete annotations," in *Proceedings of the 22nd International Conference on Computational Linguistics*, 2008.

[33] A. Neelakantan and M. Collins, "Learning dictionaries for named entity recognition using minimal supervision," in *Proceedings of the 14th European Chapter of the Association for Computational Linguistics*, 2014, pp. 452–461.

[34] G. Neubig and S. Mori, "Word-based partial annotation for efficient corpus construction," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, 2010.