IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 105–119
RECEIVED 27/10/10 ACCEPTED 26/11/10 FINAL 15/02/11

Accelerating the Sequence Annotation using Split Annotation with Active Learning

DITTAYA WANVARIE, HIROYA TAKAMURA, AND MANABU OKUMURA

Tokyo Institute of Technology, Japan

ABSTRACT

Active learning succeeds in reducing the size of labeled corpus while maintaining the high accuracy. However, active learning requires several iterations of the tagger training, which will not be practical when the training of an iteration takes long time. In this paper, we propose to simplify the all-entity labeling task by splitting the task into a set of single-entity labeling subtasks. After all entity types are labeled, we merge the data sets into an all-entity corpus and train the final tagger using the merged set. The proposed method achieved the competitive F_1 to the multi-entity learning but required much less computational time on the CoNLL chunking and named entity recognition data sets.

1 INTRODUCTION

Part-of-speech tagging, text chunking, named entity recognition, and a number of tasks in natural language processing are formulated as a sequence labeling task. Sequence labeling is a kind of classification tasks that predicts an output label for each of the corresponding input token in the sequence. Sequence labeling is also a structured labeling task which has a special property that an output label does not depend only on the input sequence, but also on the other output labels. These dependencies among output labels slow the training of the classifier, and are troublesome for an annotator. Although the labeling of a structured output corpus consumes much of human time and effort, a considerably large size of corpus is still necessary in order to train a precise classifier for a task.

Active learning is proposed to reduce the labeling cost in numerous tasks including sequence labeling [6, 11]. The intuition behind it is that only the informative samples are sufficient for the training in order to achieve high accuracy. Hence, it can reduce the annotation cost from the whole corpus to a set of informative samples in the corpus. The intuition of active learning can also be applied to a substructure level, i.e. only some substructures in the whole structure are informative. Tomanek and Hahn proposed to manually label only the informative subsequences, and automatically label the uninformative parts in the sequence [11]. Wanvarie et al. also proposed similar idea to [11], but they re-estimate the labels of uninformative parts in the training without explicitly labeling them [16]. In this paper, we adopt the method in [16] since it can achieve the similar F_1 to the supervised learning while the method in [11] cannot.

In an active learning framework, the informative set of samples are iteratively extracted from the corpus using the tagger trained on the previously labeled data. Therefore, an annotator has to wait for the training to complete before he/she can start labeling for the next iteration. If the training of the tagger takes long time, the framework will be less practical. One of the factors which causes the long training time is the number of the possible output labels. Although the conditional random fields (CRFs) can take the output labels into account in the training, the number of class labels increases and requires long training time. Cohn et al. proposed a CRFs training technique which simplifies the multi-class classification to a set of binary classifications [1]. Their method succeeded in reducing the training time of CRFs, but still required a fully labeled corpus. Here, we adopt their idea to simplify the labeling task into a set of entity labeling tasks in order to reduce the training time. For example, the labeling of a corpus consisting of 4 entity types will be split into 4 labeling subtasks. Each subtask takes only a single entity type into account.

The rest of this paper starts from the description of conditional random fields in Section 3. We summarize the active learning framework adopted in this paper and the proposed split labeling in Section 4. Section 5 contains the comparison between the proposed split labeling and the conventional all-entity labeling. Finally, we summarize the contribution of this paper and discuss the future work in Section 6.

2 Related Work

Obtaining partially labeled data is easy in many situations. For example, we can exploit the keyword link in Wikipedia text for word boundary information without any human labeling effort. In the domain adaptation task, Tsuboi et al. showed that the training using partially labeled corpus, augmented with the fully labeled source domain, achieved higher accuracy than the training with the source domain [13]. Culotta and McCallum proposed an annotation framework for an information extraction task in [2], which allows the partial annotation of the document.

Tomanek and Hahn proposed a semi-supervised active learning framework for sequence labeling which requires only informative tokens to be manually labeled [11]. Wanvarie et al. also proposed a similar system in [16, 17], but their system does not require any explicit annotation on uninformative tokens. However, all of the systems in [11, 16, 17] employ the multi-entity CRFs training which we will show later in the experiment section that the multi-entity CRFs requires long training time.

Standard L-BFGS optimizer [4] for CRFs is slow due to the full gradient computation, making the active learning impractical if an annotator has a long waiting time between the labeling iterations. In order to accelerate the training of CRFs, several optimization techniques were proposed such as stochastic gradient descent [15]. Apart from the engineering of the optimization itself, Cohn et el. proposed to simplify the multi-class learning task into a set of binary classification subtasks using error-correcting codes [1]. Their proposed method can reduce both of the training time and memory, with a slight decrease in the accuracy. Tsuruoka et al. proposed a sentence selection technique for sparse corpus annotation using active learning [14]. They also succeeded in extracting almost all of the sequences that contain the target entity within a small amount of CPU time. However, they did not employ the partial annotation. Therefore, they have to label the whole corpus in an all-entity labeling task.

3 CONDITIONAL RANDOM FIELDS (CRFS)

Given that there are an input sequence $\mathbf{x} = (x_1, ..., x_T) \in \mathbf{X}$ composed of T tokens and the corresponding output sequence $\mathbf{y} = (y_1, ..., y_T) \in \mathbf{Y}$, the conventional CRFs proposed by [3] model the probability of $\mathbf{y}$ using the following probability:

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

Z is the normalizing factor:

$$Z_{\theta,\mathbf{x},\mathbf{Y}} = \sum_{\mathbf{y}\in\mathbf{Y}} e^{\theta\cdot\mathbf{\Phi}(\mathbf{x},\mathbf{y})} \ , \tag{2}$$

which can be computed efficiently using dynamic programming. The feature function $\mathit{\Phi}$ is a mapping from $\mathbf{x}$ and $\mathbf{y}$ to a real value. Supposing that we have N training sequences and d features ($\mathbf{\Phi} = (\mathit{\Phi}_1, ..., \mathit{\Phi}_d)$), we will learn the model parameters $\theta = (\theta_1, ..., \theta_d)$, by maximizing the log likelihood of the output sequences given the input sequences in the training data:

$$\max LL(\theta) = \max \sum_{n=1}^{N} ln(P_\theta(\mathbf{y}^{(n)}|\mathbf{x}^{(n)})) \ . \tag{3}$$

We employ L-BFGS[4] with parallelized gradient computation to optimize θ in (3).

Since the sequence probability in (1) of the objective function in (3) requires a sequence to be fully labeled, we re-define the objective function for partially labeled sequences following [13] to:

$$\max LL(\theta) = \max \sum_{n=1}^{N} \ln P_\theta(\mathbf{Y}_{\mathbf{L}^{(n)}}|\mathbf{x}^{(n)}) \ . \tag{4}$$

$\mathbf{Y_L}$ is a set of all $\mathbf{y}$ consistent with the partially labeled sequence $\mathbf{x}$. The probability of $\mathbf{Y_L}$ is modified from (1) to

$$P_\theta(\mathbf{Y_L}|\mathbf{x}) = \sum_{\mathbf{y}\in\mathbf{Y_L}} P_\theta(\mathbf{y}|\mathbf{x}) \ . \tag{5}$$

4 ACTIVE LEARNING FRAMEWORK

The outline of our labeling framework is shown in Fig.1. The active learning is mostly similar to the framework in [16] augmented with the split labeling. In an iteration of each subtask, the system tries to find a set of informative tokens and ask an annotator to label them. The labeling will stop when the stopping criterion are satisfied. After the annotation of all subtasks has been finished, the partially labeled corpora from all subtasks are merged into a single multi-entity corpus for the training of the final tagger.

```
$S_{s,t}$ : $\{(\mathbf{x}, \mathbf{y_L})\}$ is a set of all training sequences with current annotation at iteration $t$ of subtask $s$
$S_{sel}$ is a set of informative tokens
$x$ is an input token
for each $s$ do
  $curmodel \leftarrow train(S_{s,1})$ {Initial training}
  repeat
    $S_{sel} \leftarrow Q_{\text{tok}}(curmodel, S_{s,t})$ {Selecting $q$ tokens (at most)}
    for $x \in S_{sel}$ do
      $S_{s,t+1} \leftarrow update(S_{s,t}, x, label(x))$ {Annotation}
    end for
    $curmodel \leftarrow train(S_{s,t+1})$ {Training}
  until ($|S_{sel}| < q$) and ($\kappa(S_{s,t}, S_{s,t+1}) > stop$) {Stopping criterion}
  $S_{s,final} \leftarrow S_{s,t+1}$
end for
$S_{final} \leftarrow merge(S_{s,final})$ {Merge training sets}
$finalmodel \leftarrow train(S_{final})$
```

Fig. 1: Active labeling framework

4.1 *Query and Labeling Strategy*

The success key of active learning in reducing the annotation cost relies on the ability to select a small set of highly *informative* labeling candidates. There are various definitions of *informative* candidates such as the prediction confidence or the information gain[6]. In this paper, we define the informativeness of a candidate by its prediction confidence owing to its simplicity. There are also several definitions of the prediction confidence itself. We follow [11, 16, 17] to define the prediction confidence using the marginal probability:

$$P_\theta(y_j = y'|\mathbf{x}) = \frac{\alpha_j(y'|\mathbf{x}) \cdot \beta_j(y'|\mathbf{x})}{Z_\theta(\mathbf{x}, \mathbf{Y})}$$

α and β are the prefix and the suffix probabilities. When the marginal probability of a token is lower than the preset threshold, we regard the token as *informative* and ask an annotator to label the token.

After a token is labeled, the marginal probability of its neighboring tokens will change and may exceeds the threshold. Thus, these tokens will require no labeling effort. The idea is called correction propagation in [2], and probability re-estimation in [16, 17]. We also employ this idea following [16] by labeling only one token per sequence per iteration, and provide at most q sequences to an annotator in each iteration.

From the objective function in (4), we cannot benefit from the entirely unlabeled sequences. On the contrary, adding unlabeled sequences will slow the optimization process. Therefore, we will train the tagger in a pass using only the fully labeled, and partially labeled sequences.

4.2 *Stopping criterion*

The stopping criterion is also an important key of active learning to reduce the annotation cost by stopping the learning when it converges. Since the labeling is done in token unit, the learning can simply stop after there is no informative token left in the corpus. However, Wanvarie et al. showed in [17] that adding a few new informative tokens does not help improving the F_1 but just wastes the training time.

We employ the stopping criterion in [16], which is described as follows. Firstly, we predict the output of all training sequences, both labeled, partially labeled, and unlabeled using the model in each iteration. Note that the output of a labeled token is exact, and always correct. Then, we measure the similarity between the prediction of the models from two consecutive iterations using Kappa statistic. The learning can be stopped if there are few differences between the prediction. We found that the usual $\kappa = 0.99$ is not sufficient to achieve high accuracy. κ is empirically tuned in the experiments. When the Kappa statistic exceeds the threshold, $\kappa = 0.9999$, the learning will stop.

4.3 *Split Labeling*

The original corpus contains various types of entity tokens. We split the labeling task into a set of single-entity labeling tasks. The labeling in each subtask is also partial labeling. The partially labeled corpora from all subtasks are merged to build the final corpus, which is still partially labeled. The final tagger is trained on the all-entity corpus using CRFs described in section 3.

We assume that there is no ambiguity from human labeling. Therefore, a token which is labeled as an entity type in a subtask will be labeled as a non-entity type in the other subtasks. In contrast, a token labeled as a non-entity type in a subtask may be a real non-entity, or an entity of the other types. A token is regarded as a non-entity token if and only if it is labeled as a non-entity type in all subtasks. Otherwise, the token is left unlabeled in the merged corpus.

However, a few non-entity tokens are labeled in all subtasks. Therefore, the merged corpus will contain mostly entity tokens, lacking of non-entity tokens, and is not appropriate for the training. We propose to label all of the unlabeled tokens by the model prediction of all subtasks in order to retrieve the non-entity tokens. However, there may be conflicts among the model predictions. In such cases, we leave the tokens unlabeled.

5 Experiments and Discussion

5.1 *Experimental Settings*

Table 1: Data statistics

Data set	#Sequences	#Tokens	#Entity types	#Entity tokens
CoNLL2000				
training	8936	211727	11	183825
test	2012	47377	10	41197
CoNLL2003				
training	14987	204567	4	34043
test	3684	46666	4	8112

We evaluated the proposed method on CoNLL data sets; the chunking task from [9] and the named entity recognition task from [10]. A sentence is represented as a sequence, while a word is represented as a token. The output label for chunking is in IOB2 format [5], while the output label for named entity recognition is in IOB format [8]. Example of each labeling

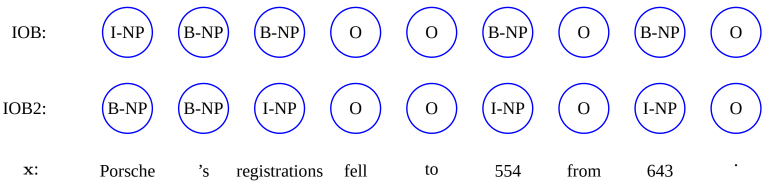


Fig. 2: Labeling example

format is shown in Figure 2. The statistics of each data set is summarized in Table 1. We used the same features described in [16, 17], which are summarized as follows.

The variables w_i and lw_i are the word and its lowercase at i distance from the current word in a sentence. p_i is the part-of-speech (POS) of w_i, which is provided in the data set. c_i is a chunk type which is provided in CoNLL2003 data set, e.g. NP chunk. wtp_i is a set of orthographic features of a word such as containing punctuations. $pw[c]_i$ and $sw[c]_i$ are c-character prefix and suffix of word w_i, e.g. 3-character prefix of the word *American* is *Ame*. y_i is an output label of w_i. Each feature template is shown in a bracket. All templates are augmented with y_i. The subscript and superscript indicate the running index of the word position. For example, $[w_i]_{i=-1}^{i=1}$ refers to three templates, w_{i-1}, w_i, and w_{i+1}.

– CoNLL2000:
  - $[w_i]_{i=-2}^{i=2}$, $[w_i, w_{i+1}]_{i=-1}^{i=0}$, $[p_i]_{i=-2}^{i=2}$, $[p_i, p_{i+1}]_{i=-2}^{i=1}$, $[p_i, p_{i+1}, p_{i+2}]_{i=-2}^{i=0}$, $[y_{i-1}]$

– CoNLL2003:
  - $[w_i]_{i=-1}^{i=1}$, $[w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1}]$, $[w_{i+1}, w_{i+2}, w_{i+3}, w_{i+4}]$
  - $[p_i]_{i=-2}^{i=2}$, $[p_i, p_{i+1}]_{i=-2}^{i=1}$, $[p_{i-1}, p_i, p_{i+1}]$, $[lw_i]_{i=-2}^{i=2}$, $[lw_i, lw_{i+1}]_{i=-2}^{i+1}$, $[wtp_i]_{i=-1}^{i=1}$
  - $[c_i]_{i=-2}^{i=2}$, $[c_i, c_{i+1}]_{i=-2}^{i=1}$, $[c_{i-1}, c_i, c_{i+1}]$
  - $[pw2_i]_{i=-1}^{i=1}$, $[pw3_i]_{i=-1}^{i=1}$, $[sw2_i]_{i=-1}^{i=1}$, $[sw3_i]_{i=-1}^{i=1}$, $[y_{i-1}]$

There are two evaluation criteria; the accuracy which is measured by F_1 using CoNLL evaluation [9], and the annotation cost. The annotation cost is also evaluated in two aspects; the number of manually labeled tokens, and the computational time in seconds.[1] We did not measure the actual annotation time by human annotators but only simulated the human annotation using the gold standard corpus.

The initial training set contains the 47 longest sequences from the training corpus. All tokens in the initial set are manually labeled. The system will provide at most new 500 informative tokens per iteration to an annotator. Although large query size requires more labeled tokens and more annotation time, a few new tokens cannot contribute much to the accuracy improvement but just wastes the computational time. Our objective is to achieve the comparable F_1 to the supervised learning, while

[1] We conducted all experiments on a Xeon 3.0GHz machine. Our CRFs implementation is in C. The optimization of CRFs is parallelized gradient computation of L-BFGS using 4 cpus.

requiring less number of labeled tokens with reasonable training time. Therefore, we should also balance the actual labeling time and the tagger training time. If the tagger training time is approximately 10-15 minutes, the suitable labeling time might be 1-2 hours. If the actual annotation time per token is few seconds [7, 12], the actual labeling time of 500 tokens will be approximately an hour and a half.

5.2 *Baseline Systems*

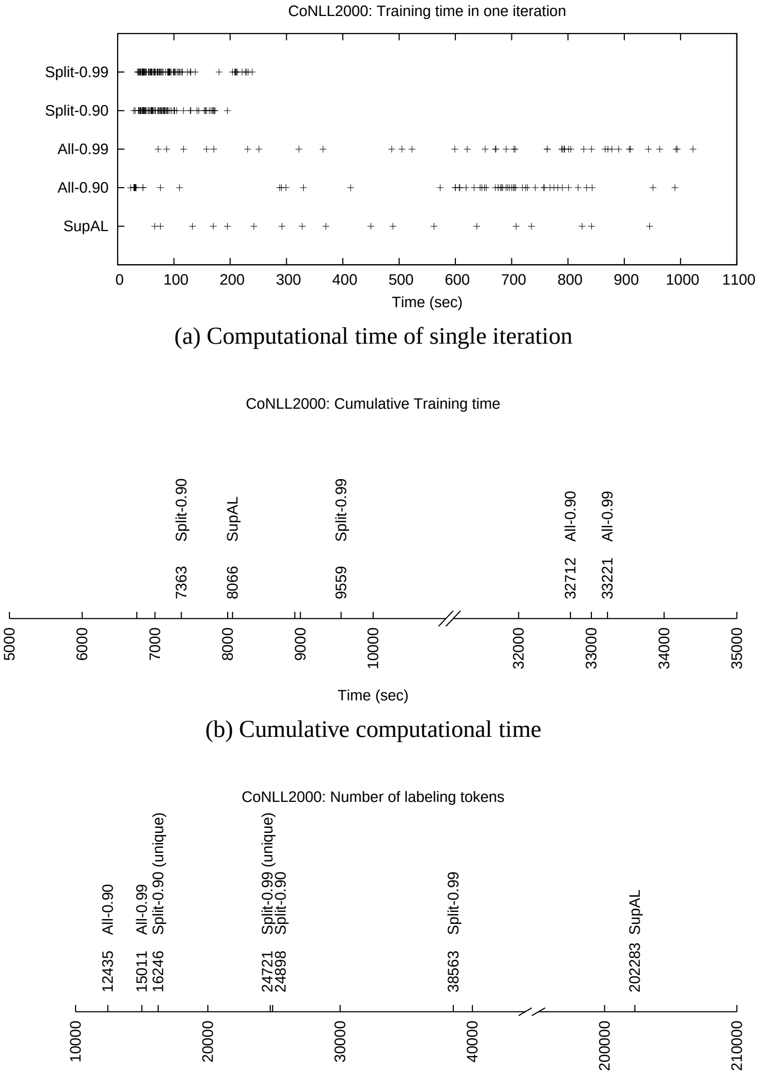
The baselines are all-entity labeling approaches. The first baseline is a supervised active learning system (*SupAL*). In each iteration, an annotator will label all tokens from a set of 500 sequences with the lowest sequence probabilities. In the following experiments, we give a bias to *SupAL* result by reporting the annotation cost when its F_1 reaches the supervised F_1 level. Note that in the real labeling situation, we do not know the real achievement of F_1 in advance. The other two baselines are partial annotation systems using all-entity training (*All*), with the confidence threshold at 0.90, and at 0.99.

5.3 *Result and Discussion*

The CPU time per iteration, which is the time an annotator has to wait before start labeling the next iteration, is shown in Figure (3a) and Figure (4a). The CPU time of all settings continued increasing when new labeled tokens were added to the training set. The CPU time in an iteration consists of the time for tagger training, token selection, and evaluating the stopping criterion. Most of the CPU time devotes to the tagger training.

In late iterations of *SupAL* and *All*, an annotator had to wait for more than 10 minutes in CoNLL2000 labeling, and more than 40 minutes in CoNLL2003 labeling. In contrast, the proposed *Split* approach reduced a half of the waiting time in both tasks. There were a few early iterations which require more than an hour in the training of CoNLL2003 experiments, when using the partially labeled training sequences. Note that most of the iterations require less than 40 minutes in the training. We also found similar phenomena in the split labeling but with much smaller amount of training time. We argue that the tagger was uncertain and might incorrectly estimate the output of unlabeled tokens, which produced strange label distribution, resulting in the long training time.

Figure (3b) and Figure (4b) show the cumulative training time of each system. While the split labeling required more training iterations than the all-entity labeling, the cumulative training time was much less.

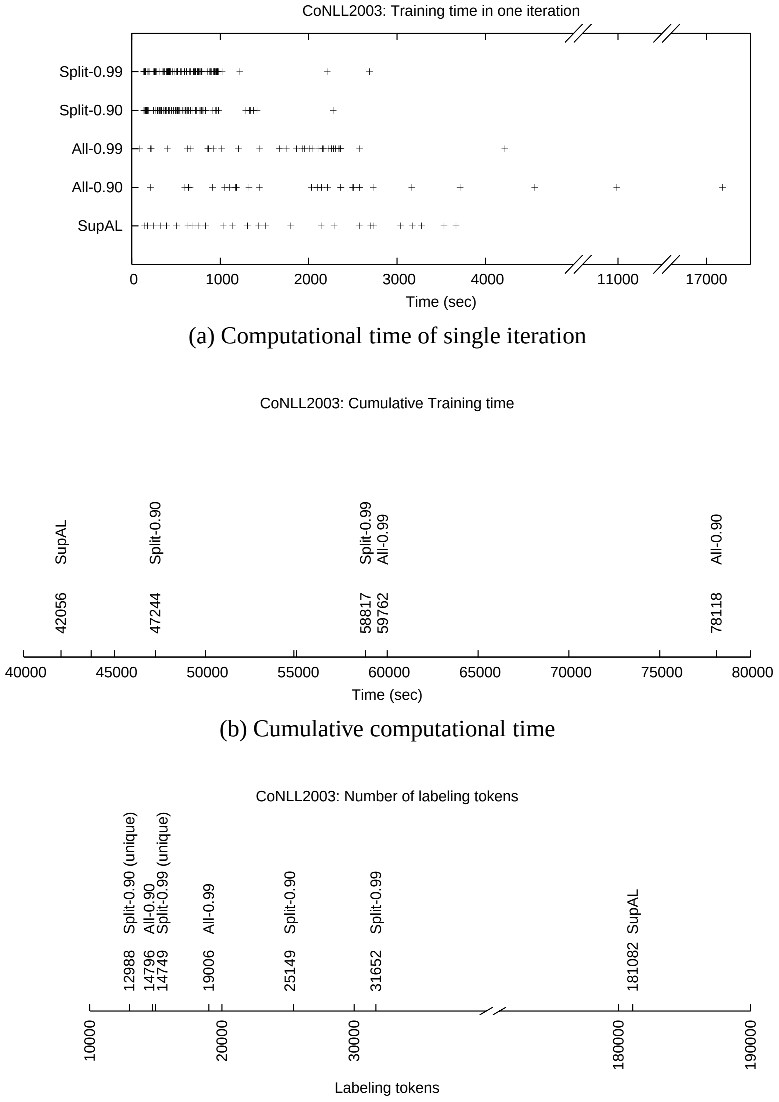


(a) Computational time of single iteration

(b) Cumulative computational time

(c) Number of labeling tokens

Fig. 3: CoNLL2000



(a) Computational time of single iteration

(b) Cumulative computational time

(c) Number of labeling tokens

Fig. 4: CoNLL2003

Figure (3c) and Figure (4c) show the number of labeled tokens in each data set. We reported the number of unique actions in split labeling using *Split*, and the number of uniquely labeled tokens using *Split (unique)*. With split labeling, a token may be labeled in several subtasks, which results in a number of labeling actions. However, we argue that labeling a single entity type is easy since the number of possible outputs in the candidate list is much less than the number in the all-entity labeling. Moreover, we can parallelize the labeling task by asking a group of annotators to label all subtasks at the same time, which can further accelerate the labeling. Another possible labeling setting is to ask the annotator to find the correct label, but strictly train the model in binary ways.

Table 2: F_1 of each system

Systems	CoNLL2000	CoNLL2003
SupAL	93.42	81.49
All-0.90	93.41	81.21
All-0.99	93.46	81.33
Split-0.90	92.98	80.46
Split-0.99	93.14	81.11

Finally, we compared F_1 of the proposed method with the baselines in Table 2. The proposed method achieved the competitive F_1 to the full labeling settings. The slight reduction of F_1 from the all-entity labeling may due to the lack of inter-entity information since the framework does not distinguish the non-entity and non-target-entity from each other in the sampling of the split subtasks. Another reason is the rare entity samples since the tagger failed to extract sufficient number of such tokens for training.

6 Conclusion and Future Work

In this paper, we have proposed a sequence annotation framework which achieved the competitive accuracy to the supervised system while required much less labeled tokens. The proposed framework also required reasonable training time compared to the exhaustive time of the all-entity labeling framework in the previous work since it could efficiently select the informative tokens in less computational time.

We may be able to further reduce the training time using faster CRFs optimization techniques such as stochastic gradient descent [15], multiclass training [1]. Other learning algorithms such as perceptron, support vector machines are also applicable to our framework. However, we need to re-define the confidence measurement and the optimization for partially labeled sequences.

ACKNOWLEDGMENTS We thank the reviewers for valuable comments and suggestions. The first author is supported from the Thai government scholarship.

REFERENCES

1. Cohn, T., Smith, A., Osborne, M.: Scaling conditional random fields using error-correcting codes. In: ACL ’05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 10–17. Association for Computational Linguistics, Morristown, NJ, USA (2005)
2. Culotta, A., McCallum, A.: Reducing labeling effort for structured prediction tasks. In: AAAI’05: Proceedings of the 20th national conference on Artificial intelligence. pp. 746–751. AAAI Press (2005)
3. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML ’01: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
4. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Math. Program. 45(3), 503–528 (1989)
5. Sang, E.F.T.K., Veenstra, J.: Representing text chunks. In: Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics. pp. 173–179. Association for Computational Linguistics, Morristown, NJ, USA (1999)
6. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: EMNLP ’08: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1070–1079. Association for Computational Linguistics, Morristown, NJ, USA (2008)
7. Settles, B., Craven, M., Friedland, L.: Active learning with real annotation costs. In: Proceedings of the NIPS Workshop on Cost-Sensitive Learning, 2008 (2008)
8. Tjong Kim Sang, E.F.: Memory-based named entity recognition. In: COLING-02: proceedings of the 6th conference on Natural language learning. pp. 1–4. Association for Computational Linguistics, Morristown, NJ, USA (2002)

9. Tjong Kim Sang, E.F., Buchholz, S.: Introduction to the conll-2000 shared task: chunking. In: Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning. pp. 127–132. Association for Computational Linguistics, Morristown, NJ, USA (2000)
10. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Daelemans, W., Osborne, M. (eds.) Proceedings of CoNLL-2003. pp. 142–147. Edmonton, Canada (2003)
11. Tomanek, K., Hahn, U.: Semi-supervised active learning for sequence labeling. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 1039–1047. Association for Computational Linguistics, Suntec, Singapore (August 2009), http://www.aclweb.org/anthology/P/P09/P09-1117
12. Tomanek, K., Hahn, U., Lohmann, S., Ziegler, J.: A cognitive cost model of annotations based on eye-tracking data. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1158–1167. Association for Computational Linguistics, Uppsala, Sweden (July 2010), http://www.aclweb.org/anthology/P10-1118
13. Tsuboi, Y., Kashima, H., Oda, H., Mori, S., Matsumoto, Y.: Training conditional random fields using incomplete annotations. In: COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics. pp. 897–904. Association for Computational Linguistics, Morristown, NJ, USA (2008)
14. Tsuruoka, Y., Tsujii, J., Ananiadou, S.: Accelerating the annotation of sparse named entities by dynamic sentence selection. In: BioNLP '08: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing. pp. 30–37. Association for Computational Linguistics, Morristown, NJ, USA (2008)
15. Vishwanathan, S.V.N., Schraudolph, N.N., Schmidt, M.W., Murphy, K.P.: Accelerated training of conditional random fields with stochastic gradient methods. In: ICML '06: Proceedings of the 23rd international conference on Machine learning. pp. 969–976. ACM, New York, NY, USA (2006)
16. Wanvarie, D., Takamura, H., Okumura, M.: Active learning with subsequence sampling strategy for sequence labeling tasks., submitted to Journal of Natural Language Processing, Japan
17. Wanvarie, D., Takamura, H., Okumura, M.: Active learning with partially annotated sequence. Tech. rep., Special Interest Group of Natural Language Processing, Information Processing of Japan (September 2010)

DITTAYA WANVARIE
DEPARTMENT OF COMPUTATIONAL INTELLIGENCE AND
SYSTEMS SCIENCE,
TOKYO INSTITUTE OF TECHNOLOGY,
4259 NAGATSUTA-CHO, MIDORI-KU, YOKOHAMA CITY, JAPAN
E-MAIL: <DITTAYA@LR.PI.TITECH.AC.JP>

HIROYA TAKAMURA
PRECISION AND INTELLIGENCE LABORATORY,
TOKYO INSTITUTE OF TECHNOLOGY,
4259 NAGATSUTA-CHO, MIDORI-KU, YOKOHAMA CITY, JAPAN
E-MAIL: <TAKAMURA@PI.TITECH.AC.JP>

MANABU OKUMURA
PRECISION AND INTELLIGENCE LABORATORY,
TOKYO INSTITUTE OF TECHNOLOGY,
4259 NAGATSUTA-CHO, MIDORI-KU, YOKOHAMA CITY, JAPAN
E-MAIL: <OKU@PI.TITECH.AC.JP>