Vol.2010-NL-198 No.4
2010/9/17

Active Learning with Partially Annotated Sequence

Dittaya Wanvarie,[†1] Hiroya Takamura[†2]
and Manabu Okumura[†2]

We propose an active learning framework which requires human annotation only in the ambiguous parts of the sequence. In each iteration of active learning, a set of tokens from the ambiguous parts are manually labeled while the other tokens are left unannotated. Our proposed method is superior to the method where unambiguous tokens are automatically labeled. We evaluate our proposed framework on chunking and named entity recognition data provided by CoNLL. Experimental results show that our proposed framework outperforms the previous work using automatically labeled tokens, and almost reaches the supervised F_1 with 6.37% and 8.59% of tokens being manually labeled, respectively.

1. Introduction

The supervised learning is well-known to produce a highly accurate result with the traded-off expense of data annotation. In some tasks such as document polarity classification, data annotation is not difficult and does not require an expert to perform the job. However, tasks such as part-of-speech tagging are complex and require an expert to label the training data. In such cases, the cost of annotation is extremely high and may not be affordable.

Active learning[4] is proposed to reduce the annotation cost from conventional supervised approaches by elaborately selecting and annotating a set of informative samples for training. The key of success in annotation cost reduction of active learning relies on a query strategy which will return a small set of highly informative samples. We can train a model only on the informative set and achieve the comparable performance to the model trained on the whole training set.

†1 Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology
†2 Precision and Intelligence Laboratory, Tokyo Institute of Technology

Sequence labeling is a kind of structural learning where a sequence structure constructed of tokens. The objective of the task is to find a label sequence of an input sequence where each output label correspond to an input token. The task is not trivial since each output does not only depend on the input sequence but also depends on output of the other input tokens. Its complex structure is also the reason of high annotation cost.

We use a classifier, specifically called an annotator or tagger for a sequence labeling task, to predict a label output of each token in the sequence. Each label is usually predicted with different confidence level. In supervised learning, an annotator will manually label all tokens in a sequence regardless of its prediction confidence. However, we may reduce the annotation by only labeling tokens with low confidence. The method turns to be semi-supervised learning.

In this paper, we adopt the conditional random fields (CRFs) which is the state-of-the-art method in sequence labeling as our tagger. However, the conventional CRFs requires the training sequence to be fully labeled.[7] We cannot directly adopt the conventional CRFs if we only label the low confidence tokens. One solution is to fill the partially labeled sequences with the model prediction,[16] but the model prediction may not be accurate, especially in early iterations. Adding such incorrectly labeled tokens into the training set will result in poor tagger. We propose an active learning framework on partially labeled sequences. We can train the tagger in each iteration using the modified the CRFs which can estimate the model parameters from partially labeled sequences.[18] As the model become more accurate in later iterations, the prediction errors in early iteration may be automatically fixed.

The organization of the rest of the paper is as follows. We start discussing related work to our proposed framework in section 2. Section 3 is devoted to the parameter estimation of conditional random fields (CRFs) which is the tagger adopted in our framework. In section 4, we describe our proposed annotation settings in detail. Section 5 contains the experiments, discussion and analysis of the result. Finally, we conclude our contribution and discuss the future work in section 6.

2. Related work

Semi-supervised learning benefit from the high accurate but costly labeled data, and noisy but cheap unlabeled data. Self-training may be the simplest semi-supervised technique where we create the pseudo-labeled data from unlabeled data using the model trained on the labeled set. However, we can achieve little improvement by this method especially when we have a small amount of labeled data. Since we assume that the unlabeled set has the same distribution as the labeled set, we can learn little new information from the unlabeled data. Actually, we can apply an unsupervised learning method to train a tagger from the unlabeled set. When we have both labeled and unlabeled data, we can separately train a tagger from each type of data and combine them together using ensemble methods. We can also train a tagger using both types of data at the same time but assigning each type of data with its appropriate parameters.[1),13)] Another approach is to learn the general concept from one type of data and and refine the tagger using labeled data.[9)]

The difficulty of labeling is often caused by the dependency between substructures. Obtaining partially labeled data in structural learning may be easier than obtaining the fully labeled data. Although we know only the information from a part of a sample, we can find the information from the other parts in another sample. Therefore, we can also train a model on partially labeled data.[10),12)] The idea is also applicable to the domain adaptation task where different domains share some general information. We only need to provide the domain specific information from the target domain data, and gather the general information from the source domain data.[18)]

The current model may already be able to predict the correct output of some samples. In other words, these samples contain little new information for the training. We define such samples as *uninformative* smaples and may limit the annotation effort on these samples.[6)] We can measure the informativeness of a sample in several ways. Settles and Craven have analyzed several active learning strategies for fully labeled sequence labeling task[11)]. For a structural sample which consists of substructures, the model may already be able to predict only a part of the sample. Hence, we can further reduce the annotation cost by changing the labeling from the whole structure labeling to substructure labeling.[16)]

Bootstrapping is closely related to active learning in the way that new samples are selected and added to the training set in each iteration. In contrast to active learning, bootstrapping requires no human annotation effort. Therefore, bootstrapping strategy will select new samples whose prediction confidence is rather high to avoid labeling errors.[3),20)] Although bootstrapping requires less annotation cost than active learning, the performance of bootstrapping is still far poorer than that of the active learning, which exploits expensive annotated data.

3. Conditional Random Fields

The objective of the sequential labeling task is to find an output label sequence $\mathbf{y} = (y_1, ..., y_T) \in \mathbf{Y}$ of the input sequence $\mathbf{x} = (x_1, ..., x_T) \in \mathbf{X}$. T is the length of the sequence. $\mathbf{X}$ and $\mathbf{Y}$ are the sets of all possible input and output sequences, respectively. We will learn the mapping: $\mathbf{X} \rightarrow \mathbf{Y}$.

The conventional CRFs proposed in 7) model the conditional probability of output label sequence $\mathbf{y}$ given input sequence $\mathbf{x}$ as

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

where $\mathbf{\Phi}(\mathbf{x},\mathbf{y}) : \mathbf{X} \times \mathbf{Y} \rightarrow \mathbb{R}^d$ is a function from a pair of input sequence $\mathbf{x}$ and output sequence $\mathbf{y}$ to a feature vector of d dimensions. $Z_{\theta,\mathbf{x},\mathbf{Y}}$ is the normalizing factor defined by

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

Let α_j be the probability of the prefix sequence until position j, called the forward probability:

$$\alpha_j(y') = \sum_{y''} \left(\alpha_{j-1}(y'') p_t(y'', y') p_s(y')\right) ,$$
$$\alpha_1(y') = p_s(y') \ .$$

Let β_j be the probability of the suffix sequence from position j, called the backward probability:

$$\beta_j(y') = \sum_{y''} \left(p_t(y', y'') p_s(y'') \beta_{j+1}(y'')\right) ,$$
$$\beta_T(y') = 1 \ .$$

$p_t(y', y'')$ is the transition probability from label y' to label y''. $p_s(y')$ is the

output probability of label y'. We can efficiently compute $Z_{\theta,\mathbf{x},\mathbf{Y}}$ by

$$Z_{\theta,\mathbf{x},\mathbf{Y}} = \sum_{y' \in Y_1} \alpha_1(y') \cdot \beta_1(y')$$

where Y_1 is all possible labels of y_1.

$\theta \in \mathbb{R}^d$ is a set of model parameters. For a set of N training sequences $\{(\mathbf{x}^i, \mathbf{y}^i)\}$, the learning process will maximize the following log likelihood function:

$$LL(\theta) = \sum_{n=1}^{N} ln(P_\theta(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}) \ . \tag{2}$$

We can apply standard optimization techniques such as L-BFGS[8] or SGD[19] to the objective function in (2).

Given a partially labeled sequence or ambiguously labeled sequence $\mathbf{L}$, let $\mathbf{Y_L}$ be the set of all possible output sequences consistent with $\mathbf{L}$. We follow 2), 18) to estimate the probability of $\mathbf{Y_L}$ given $\mathbf{x}$ by

$$P_\theta(\mathbf{Y_L}|\mathbf{x}) = \sum_{\mathbf{y} \in \mathbf{Y_L}} P_\theta(\mathbf{y}|\mathbf{x}) \ . \tag{3}$$

Using (3), the log likelihood in (2) is modified to

$$\begin{aligned} LL(\theta) &= \sum_{n=1}^{N} \ln P_\theta(\mathbf{Y}_{\mathbf{L}^{(n)}}|\mathbf{x}^{(n)}) \\ &= \sum_{n=1}^{N} (\ln \sum_{\mathbf{y} \in \mathbf{Y}_{\mathbf{L}^{(n)}}} \frac{e^{\theta \cdot \mathbf{\Phi}(\mathbf{x}^{(n)}, \mathbf{y})}}{\sum_{\mathbf{y}' \in \mathbf{Y}} e^{\theta \cdot \Phi(\mathbf{x}^{(n)}, \mathbf{y}')}}) \\ &= \sum_{n=1}^{N} (\ln \sum_{\mathbf{y} \in \mathbf{Y}_{\mathbf{L}^{(n)}}} P_\theta(\mathbf{y}|\mathbf{x}^{(n)}) - \ln \sum_{\mathbf{y} \in \mathbf{Y}} P_\theta(\mathbf{y}|\mathbf{x}^{(n)})) \\ &= \sum_{n=1}^{N} (\ln Z_{\theta,\mathbf{x}^{(n)},\mathbf{Y}_{\mathbf{L}^{(n)}}} - \ln Z_{\theta,\mathbf{x}^{(n)},\mathbf{Y}}) \ . \end{aligned} \tag{4}$$

$\mathbf{x}^{(n)}$ and $\mathbf{L}^{(n)}$ are n^{th} input sequence and a set of all possible output sequence, respectively. $Z_{\theta,\mathbf{x}^{(n)},\mathbf{Y}_{\mathbf{L}^{(n)}}}$ can be computed by the forward-backward algorithm similar to the one used for $Z_{\theta,\mathbf{x},\mathbf{Y}}$. We then apply the standard optimization techniques to (4) as done in (2).

4. Proposed Framework

In active learning, we start labeling from the sequences with low output probability since such sequences are likely to contain more information than sequences with high output probability. However, there are several possible output sequences for an input sequence. In order to compare the probability among input sequences, we compare the probability of the Viterbi sequence of each input. The Viterbi sequence, $\hat{\mathbf{y}}$, is defined by

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \ .$$

The Viterbi sequence probability is calculated by (1). We rank sequences in the training set by its Viterbi sequence probability and extract all informative tokens from the top q sequences as the queried set for labeling. After a sequence is extracted and labeled, we will not re-label any tokens in the sequence.

When a model predicts an output of a token with low confidence, the token may contain crucial information which is previously unknown to the model. As a result, we should give the information to the model by manually labeling the token. We call such a token as *informative*. In contrast, if the current model is already able to predict output of a sample with high confidence, there is little new information that the token can contribute to the training. Therefore, we call such a token as *uninformative* and will not spend annotation cost on it.

We define the prediction confidence of each token in the Viterbi sequence by its marginal probability:

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

where α and β are the forward and backward probabilities defined in section 3. A token with the confidence less than the threshold, δ, is regarded as an informative token.

Tomanek and Hahn proposed to automatically label uninformative tokens by the model prediction.[16] The method is called *Se*mi-*S*upervised *A*ctive *L*earning system (*SeSAL*). However, we have discussed that the automatically predicted labels may not be correct. Instead, we propose a *Partial* annotation setting (*Partial*), where high confidence tokens are left unannotated. We can train the tagger using partially labeled sequences using the modified objective function of CRFs in (4). Since we do not explicitly label any high confidence tokens, there is a chance that the prediction errors in early iterations may be fixed after the tagger becomes more accurate in later iterations.

Table 1 Feature set

Feature type	Templates
CoNLL2000	
Word	$[w_{i-2}], [w_{i-1}], [w_i], [w_{i+1}], [w_{i+2}], [w_{i-1}, w_i], [w_i, w_{i+1}]$
	$[w_{i-2}, w_{i-1}, w_i], [w_{i-1}, w_i, w_{i+1}], [w_i, w_{i+1}, w_{i+2}]$
POS	$[p_{i-2}], [p_{i-1}], [p_i], [p_{i+1}], [p_{i+2}], [p_{i-2}, p_{i-1}], [p_{i-1}, p_i], [p_i, p_{i+1}], [p_{i+1}, p_{i+2}]$
Transition	$[y_{i-1}]$
CoNLL2003	
Unigram	$[w_{i-2}], [w_{i-1}], [w_i], [w_{i+1}], [w_{i+2}], [lw_{i-2}], [lw_{i-1}], [lw_i], [lw_{i+1}], [lw_{i+2}],$
	$[p_{i-2}], [p_{i-1}], [p_i], [p_{i+1}], [p_{i+2}], [c_{i-2}], [c_{i-1}], [c_i], [c_{i+1}], [c_{i+2}],$
	$[wtp_{i-2}], [wtp_{i-1}], [wtp_i], [wtp_{i+1}], [wtp_{i+2}]$
Bigram	$[lw_{i-2}, lw_{i-1}], [lw_{i-1}, lw_i], [lw_i, lw_{i+1}], [lw_{i+1}, lw_{i+2}],$
	$[p_{i-2}, p_{i-1}], [p_{i-1}, p_i], [p_i, p_{i+1}], [p_{i+1}, p_{i+2}], [c_{i-2}, c_{i-1}],$
	$[c_{i-1}, c_i], [c_i, c_{i+1}], [c_{i+1}, c_{i+2}]$
Trigram	$[p_{i-1}, p_i, p_{i+1}], [c_{i-1}, c_i, c_{i+1}]$
PrevWord	$[w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1}]$
NextWord	$[w_{i+1}, w_{i+2}, w_{i+3}, w_{i+4}]$
Prefix	$[pw2_{i-1}], [pw2_i], [pw2_{i+1}], [pw3_{i-1}], [pw3_i], [pw3_{i+1}]$
Suffix	$[sw2_{i-1}], [sw2_i], [sw2_{i+1}], [sw3_{i-1}], [sw3_i], [sw3_{i+1}]$

5. Experiments and Result

5.1 Data and Evaluation

We perform experiments on 2 datasets, CoNLL-2000 Chunking task[14] and CoNLL-2003 named entity recognition task[15]. The feature templates for each data set are shown in **Table 1**. Each template is shown in a bracket. w_i and lw_i is a word and lowercase word at position i in a sentence. p_i is the part-of-speech (POS) of w_i. c_i is a chunk type, e.g. an NP chunk. wtp is the word type described in **Table 2**. $pw[c]_i$ and $sw[c]_i$ are c character prefix and suffix of word w_i. y_i is an output label of w_i. Each label is either in the format of Begin-Chunk, Inside-chunk or Outside chunk, e.g. B-NP is the beginning token of an NP chunk. We analyze the proposed annotation setting using CoNLL-2000 chunking task dataset, but simplify the task to noun phrase chunking. Hence, we have 3 types of label in this simplified set, B-NP, I-NP, and O.

We analyze the performance of active learning by a learning curve between F_1 and the human annotation cost. There are several methods to measure the annotation cost. Some complex methods are suggested in 5), 17). However, we simply use the number of manually labeled tokens as the annotation cost since it is believed to be a good approximation.[10),16)]

Table 2 Word type and examples

Description	Examples
starts with capital letter	Confidence, September, But
all capital letters	PLC, GNP, A
contains both uppercase and lowercase letters	anti-American, Chancellor, Lawson
single digit	1, 2, 3
contains only digits	16, 1988, 190
contains at least two periods	U.K., A.P., F.S.B
ends with a period	Ala., p.m., vs.
contains a dash	year-ago, 1-800-453-9000, 10-fold
single character	A, a, b
contains punctuation	US$, #, ','
contains quotation	's, I'm, n't

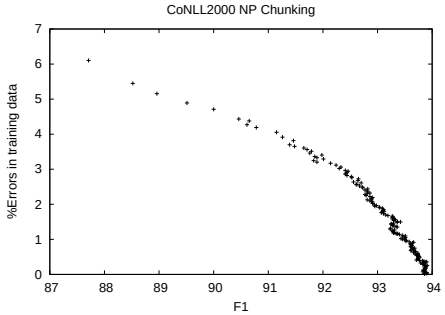

Fig. 1 Correlation between expected errors in training data and F_1

Apart from F_1, we also analyze the correctness of each model using prediction errors in the training set. If a model can precisely predict labels in the training set, it is likely that the model will also precisely predict the labels in the test set. We plot the correlation between percent of prediction errors in the training set and F_1 on the test set in **Fig. 1**. The correlation between prediction errors and F_1 is -0.99. In order to achieve the supervised F_1, the model should produce errors in the training set less than 2%.

In all experiments, we start active learning with 50 longest sequences and query for new 50 sequences in each iteration. The confidence threshold, δ, is set to 0.99

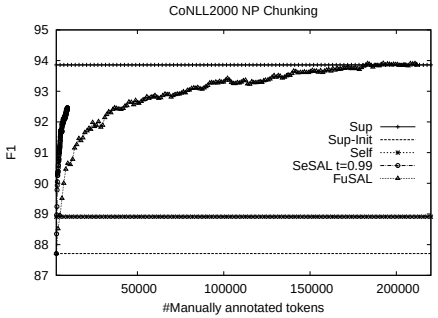

Fig. 2 F_1 and number of manually labeled tokens in the baseline systems

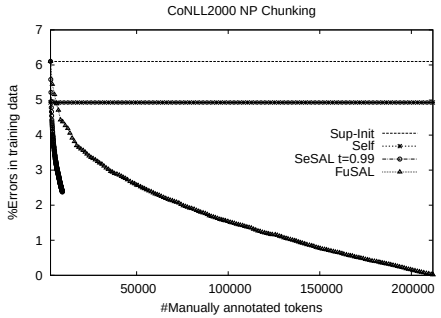

Fig. 3 Expected errors in training data of each baseline

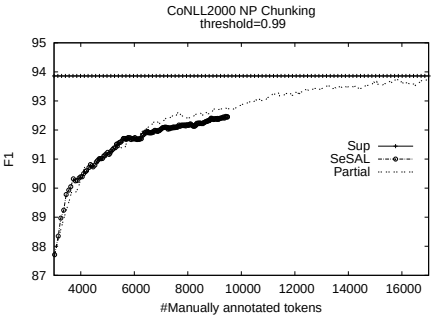

Fig. 4 F_1 of *SeSAL* and *Partial*

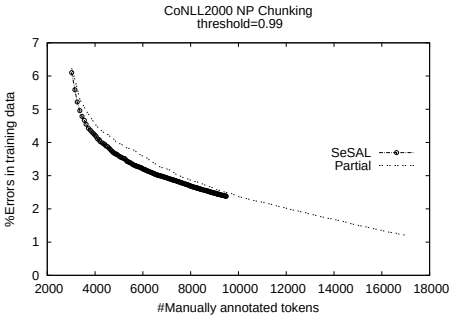

Fig. 5 Errors of *SeSAL* and *Partial*

if not explicitly specified.

5.2 Baseline systems

There are 5 baseline systems in our experiments.

Sup: The first baseline is the supervised F_1 (*Sup*) trained on the whole training set. The *Sup* baseline is the upper-bound of all systems.

Sup-Init: The second baseline is also a supervised system but is trained on only the initial set of active learning (*Sup-Init*). Therefore, the *Sup-Init* is the lower-bound baseline for all systems.

Self: The third baseline is self-training (*Self*). Only the initial set is manually labeled. We automatically label the other sequences using the model prediction.

FuSAL: The fourth baseline is a fully supervised active learning system (*FuSAL*) where all tokens in each sequence are manually labeled. Hence, after we label the whole training set, *FuSAL* become exactly the *Sup* system.

SeSAL: The last baseline is a semi-supervised active learning (*SeSAL*) proposed in 16) whose low confidence tokens are manually labeled and high confidence tokens are automatically labeled by the model prediction.

We compare each baseline system in **Fig. 2** and **Fig. 3**. Note that the F_1 of *Sup*, *Sup-Init*, and *Self* do not vary since the number of manually labeled tokens is fixed. We just draw straight lines of these settings for reference. *Self* gained slight improvement of F_1 over *Sup-Init* with no additional annotation effort. Both *FuSAL* and *SeSAL* achieved higher F_1 than *Self* with little amount of annotation cost. Although *SeSAL* achieved rather high F_1 using few labeled tokens, the highest F_1 of *SeSAL* did not reach the supervised F_1 level. On the other hand, *FuSAL* requires more tokens to be labeled but achieved the supervised F_1 using approximately 75% of labeled tokens in the training set. We can see from Fig. 3 that errors in the training data of *SeSAL* was more than 2% which then prevented the tagger from achieving the supervised F_1.

5.3 F_1 and the Annotation Cost

From **Fig. 4**, both *SeSAL* and *Partial* achieved similar F_1 if the same number of labeled tokens are provided. However, *Partial* requires more annotation cost for the while training set and achieved higher F_1 than *SeSAL*. We can also see from **Fig. 5** that the expected errors of *Partial* continued decreasing to 1.2% which is low enough to achieve the supervised F_1. We can conclude that some labeling errors in early iterations are recovered in later iterations after more tokens are manually labeled.

5.4 Effect of the Confidence Threshold

With high confidence threshold settings, the tagger becomes more reliable with the traded-off of the human annotation cost. From the result in **Fig. 6**, the curve with low threshold, $\delta = 0.60$, stopped early before reaching the supervised F_1. We argue that there are too few tokens being manually labeled resulting in many errors in the training set. When we increased the threshold, there were more tokens being labeled and the tagger achieved higher F_1. With the highest threshold, $\delta = 0.99$, the learning curve of *Partial* finally reached the supervised

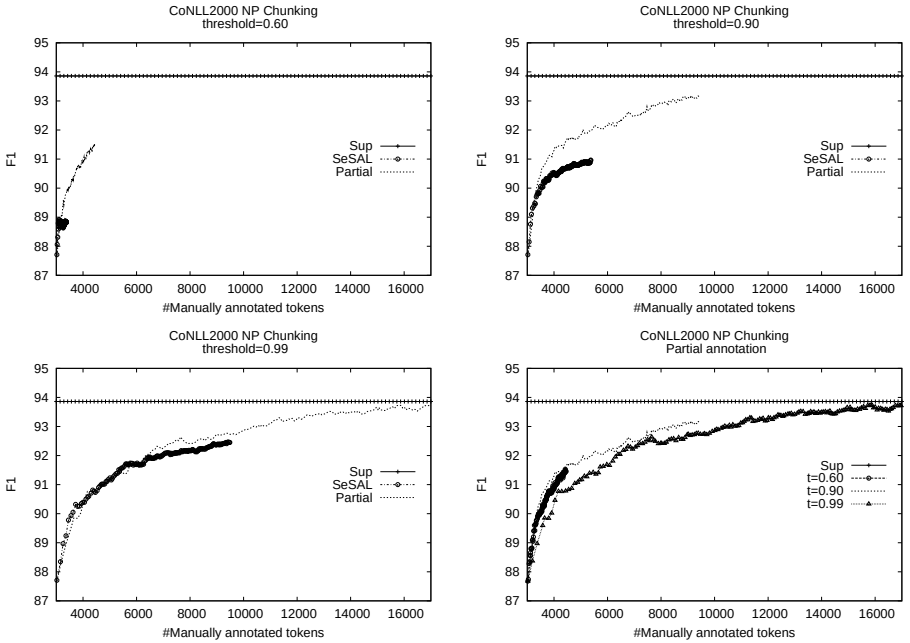

Fig. 6 Effect of threshold setting

F_1 level.

Compared to the previous work, even with the high threshold $\delta = 0.99$, the learning curve of *SeSAL* did not reach the supervised F_1. Since all tokens are explicitly labeled in *SeSAL*, the model becomes certain with few annotation cost. However, there are too many incorrectly labeled tokens included in the training set and prevent *SeSAL* from achieving the supervised F_1. In contrast, the training on partially annotated sequences requires more annotation cost but the total number of incorrectly labeled tokens is reduced at the same time. Hence, the model trained on partially annotated sequences can achieve the supervised F_1.

5.5 Result on Chunking and Named Entity Recognition

Finally, we perform experiments on full chunking and named entity recognition tasks. **Figure 7** and **Fig. 8** show that our proposed algorithm consistently outperforms *SeSAL* in both datasets. The F_1 of our proposed framework almost

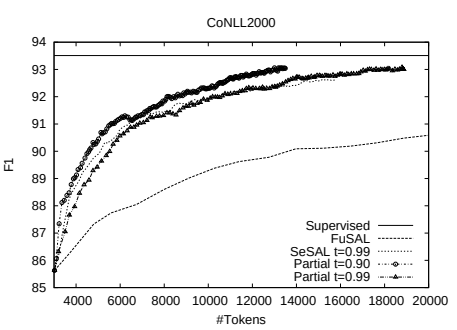

Fig. 7 Result on CoNLL chunking task

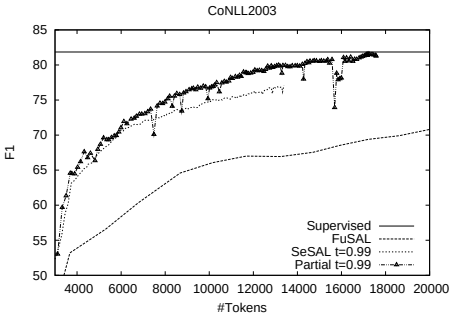

Fig. 8 Result on CoNLL Named Entity Recognition task

reached the supervised F_1 with only 5-10% of tokens being manually labeled.

6. Conclusion and Future Work

We have proposed a partial annotation setting for sequence labeling which requires few manually labeled tokens to achieve the supervised F_1. The key of our annotation is the parameter re-estimation on partially labeled sequences which can recover the prediction errors from previous iterations. We are also planning to extend our work to more complex structures such as trees or graphs based on the similar idea.

One can refine this work on the tagger itself since training CRFs is time consuming. A light-weight tagger such as perceptron might be more suitable to active learning than CRFs. Moreover, the annotation cost modeling must be more realistic, especially represents the time required in the annotation in order to adjust the appropriate query size parameter.

References

1) Ando, R.K. and Zhang, T.: A high-performance semi-supervised learning method for text chunking, *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, Association for Computational Linguistics, pp.1–9 (2005).
2) Bellare, K. and McCallum, A.: Learning extractors from unlabeled text using relevant databases, *Sixth International Workshop on Information Integration on*

the Web (2007).
3) Bellare, K. and McCallum, A.: Generalized expectation criteria for bootstrapping extractors using record-text alignment, *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, Association for Computational Linguistics, pp.131–140 (2009).
4) Cohn, D., Atlas, L. and Ladner, R.: Improving Generalization with Active Learning, *Mach. Learn.*, Vol.15, No.2, pp.201–221 (1994).
5) Culotta, A. and McCallum, A.: Reducing labeling effort for structured prediction tasks, *AAAI'05: Proceedings of the 20th national conference on Artificial intelligence*, AAAI Press, pp.746–751 (2005).
6) Dasgupta, S. and Ng, V.: Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification, *ACL-IJCNLP '09: 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: Volume 2*, Morristown, NJ, USA, Association for Computational Linguistics, pp.701–709 (2009).
7) Lafferty, J.D., McCallum, A. and Pereira, F. C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp.282–289 (2001).
8) Liu, D.C. and Nocedal, J.: On the limited memory BFGS method for large scale optimization, *Math. Program.*, Vol.45, No.3, pp.503–528 (1989).
9) Raina, R., Battle, A., Lee, H., Packer, B. and Ng, A.Y.: Self-taught learning: transfer learning from unlabeled data, *ICML '07: Proceedings of the 24th international conference on Machine learning*, New York, NY, USA, ACM, pp.759–766 (2007).
10) Sassano, M. and Kurohashi, S.: Using Smaller Constituents Rather Than Sentences in Active Learning for Japanese Dependency Parsing, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, Association for Computational Linguistics, pp.356–365 (2010).
11) Settles, B. and Craven, M.: An analysis of active learning strategies for sequence labeling tasks, *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, Association for Computational Linguistics, pp.1070–1079 (2008).
12) Spreyer, K. and Kuhn, J.: Data-driven dependency parsing of new languages using incomplete and noisy training data, *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Morristown, NJ, USA, Association for Computational Linguistics, pp.12–20 (2009).
13) Suzuki, J. and Isozaki, H.: Semi-Supervised Sequential Labeling and Segmentation Using Giga-Word Scale Unlabeled Data, *Proceedings of ACL-08: HLT*, Columbus, Ohio, Association for Computational Linguistics, pp.665–673 (2008).
14) Tjong KimSang, E.F. and Buchholz, S.: Introduction to the CoNLL-2000 shared task: chunking, *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*, Morristown, NJ, USA, Association for Computational Linguistics, pp.127–132 (2000).
15) Tjong KimSang, E.F. and DeMeulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, *Proceedings of CoNLL-2003* (Daelemans, W. and Osborne, M., eds.), Edmonton, Canada, pp. 142–147 (2003).
16) Tomanek, K. and Hahn, U.: Semi-Supervised Active Learning for Sequence Labeling, *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*, Suntec, Singapore, Association for Computational Linguistics, pp.1039–1047 (2009).
17) Tomanek, K., Hahn, U., Lohmann, S. and Ziegler, J.: A Cognitive Cost Model of Annotations Based on Eye-Tracking Data, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, Association for Computational Linguistics, pp.1158–1167 (2010).
18) Tsuboi, Y., Kashima, H., Oda, H., Mori, S. and Matsumoto, Y.: Training conditional random fields using incomplete annotations, *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, Morristown, NJ, USA, Association for Computational Linguistics, pp.897–904 (2008).
19) Vishwanathan, S. V.N., Schraudolph, N.N., Schmidt, M.W. and Murphy, K.P.: Accelerated training of conditional random fields with stochastic gradient methods, *ICML '06: Proceedings of the 23rd international conference on Machine learning*, New York, NY, USA, ACM, pp.969–976 (2006).
20) Wu, D., Lee, W.S., Ye, N. and Chieu, H.L.: Domain adaptive bootstrapping for named entity recognition, *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Morristown, NJ, USA, Association for Computational Linguistics, pp.1523–1532 (2009).