Journal of Data Intelligence, Vol. 1, No. 3 (2020) 319–332

© Rinton Press

PARTIAL ANNOTATION SCHEME FOR ACTIVE LEARNING ON NAMED ENTITY RECOGNITION TASKS

KOGA KOBAYASHI
Graduate School of Library, Information and Media Studies, University of Tsukuba
Kasuga 1-2, Tsukuba, Ibaraki 305-8550, Japan
s1921631@s.tsukuba.ac.jp

KEI WAKABAYASHI
Faculty of Library, Information and Media Studies, University of Tsukuba
Kasuga 1-2, Tsukuba, Ibaraki 305-8550, Japan
kwakaba@slis.tsukuba.ac.jp

Active learning is a promising approach to alleviate the expensive annotation cost for making training data on named entity recognition (NER) tasks. However, since existing active learning methods on NER tasks implicitly assume the full annotation scheme of which the unit of an annotation request is the whole sentence, the efficiency of the data instance selection is limited. In this paper, we propose a new active learning method based on a partial annotation scheme, which selects a part of the sentences to be annotated and asks human annotators to label a specific part of the target sentences. In the experiment, we show that the partial annotation scheme can quickly train the proposed point-wise prediction model compared to the existing active learning methods on NER tasks.

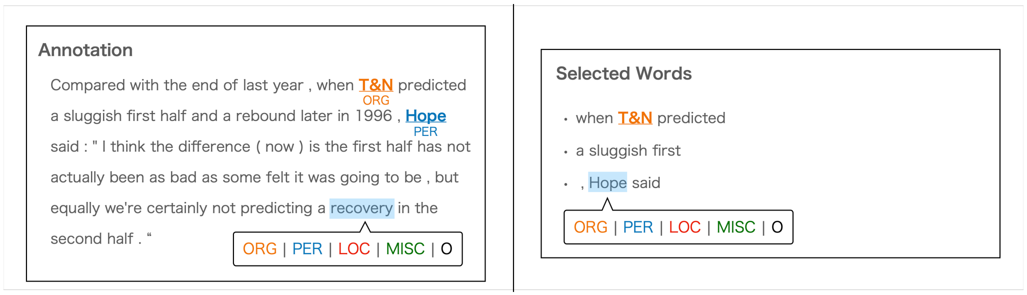
Keywords: Neural networks, Named entity recognition, Text tagging

1. Introduction

Named entity recognition (NER) is one of the fundamental processes in natural language processing that automatically extracts named entities (e.g., the name of a person, an organization, a location). NER brings a basic semantic awareness into natural language applications such as information retrieval and question answering [1]. Particularly, in recent years, a growing number of digitalized text information is available across many different specialized domains such as patents, recipes, posts in a programming forum, papers in a specific research field, etc. In this situation, there are increasing potential needs to create a custom NER model for processing named entities in each domain adequately because specialized domain has its own terminology that cannot be recognized by general NER models.

One of the primary issues in the training of a custom NER model is the cost for making training data, as known as annotation corpus. Annotation corpus for NER is a collection of sentences along with annotations of named entity tags on phrases in the sentences. These annotations need to be provided by domain experts through an annotation interface (Fig. 1 **(Left)**). Since the annotation task requires domain knowledge and time-consuming work for reading the whole sentence to find out all the named entities, the cost for making training data on NER tasks is expensive.

One of the promising approaches to reduce the annotation cost is active learning. Active



Compared with the end of last year , when T&N predicted a sluggish first half and a rebound later in 1996 , Hope said : " I think the difference (now) is the first half has not actually been as bad as some felt it was going to be , but equally we're certainly not predicting a recovery in the second half . “

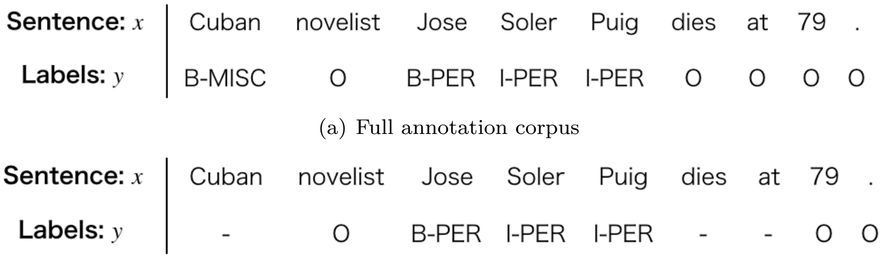
- when T&N predicted
- a sluggish first
- , Hope said

Fig. 1. **(Left)** In annotation task for NER in the full annotation scheme, human annotators are asked to check a whole sentence. **(Right)** In the partial annotation scheme, human annotators are asked to check a specific short part in sentences.

learning is a method that examines the current behavior of a machine learning model and chooses a data instance that is expected to be the most useful for training the machine learning model [2]. By controlling the annotation request order of the sentences, the performance of a machine learning model is known to be significantly improved compared with the performance of the model trained by the same number of sentences annotated in random order. Active learning methods for NER models have been proposed to date [3, 4]. However, these methods have a potential issue that may miss an opportunity to achieve better performance due to its implicit assumption on the annotation scheme; the existing active learning methods are proposed under a full annotation scheme, i.e., the algorithm chooses a sentence and asks human annotators to check the whole sentence in an annotation request. Especially when the sentence is long, the granularity of data instance that an active learning method can choose is coarse. Even if a current NER model needs to learn a specific phrase in a sentence, we have to ask human annotators to check the whole sentence that may contain terms that can be well recognized already by the current model.

To address this potential issue, we consider the partial annotation scheme that asks human annotators to check a short part of the target sentences at each request (Fig. 1 **(Right)**). Under this scheme, we propose a new active learning method that selects a part of the sentences to be annotated. The fine-grained annotation request is expected to make the annotation cost lower for achieving the same performance compared to the active learning on the full annotation scheme. The challenge of actualizing this scheme is to relax the constraint on the structure of the training data. In fact, sentence is the unit of training data instance that is assumed by the major NER models. In other words, we cannot train such models by using partially annotated sentences. The proposed method in this paper avoids this problem by adopting point-wise prediction model [5] that can be trained by using partially annotated sentences.

This paper is composed as follows. In Section 2, we introduce research on the relationship between NER and active learning and introduce some papers on applying active learning to the specific named entity recognition task. In Section 3, we describe the method of extracting named entities by point-wise prediction and applying it to active learning. In Section 4 and 5, we explain a comparison experiment between the existing method and our method after an experiment to show the effectiveness of active learning for point-wise prediction. In Section 6, we discuss the conclusions and future work.



Sentence: x	Cuban	novelist	Jose	Soler	Puig	dies	at	79	.
Labels: y	B-MISC	O	B-PER	I-PER	I-PER	O	O	O	O

(a) Full annotation corpus

Sentence: x	Cuban	novelist	Jose	Soler	Puig	dies	at	79	.
Labels: y	-	O	B-PER	I-PER	I-PER	-	-	O	O

(b) Partial annotation corpus

Fig. 2. Example of annotation corpus

2. Related Work

2.1. *Named entity recognition*

The NER tasks are typically formulated as a sequence labeling task that predicts the labels corresponding to each word. For representing the information specifying named entities, various tag formats (e.g., IOB2, IOE2, and IOBES) have been proposed. In this paper, we adopt the IOB2 format. The IOB2 format represents the role of each word by using an entity type tag with a prefix “I”, “B”, or “O”. “B” indicates “Begi” which means the first word of the named entity phrase. “I” is “Inside” which means the second or later word of the named entity phrase. “O” stands for “Other” indicating the word is not a part of named entity phrases. Fig. 2(a) shows an example of sequence labels in the IOB2 format with two entities: “Jose Soler Puig” as a person’s name (PER) and “Cuban” as a miscellaneous entity name (MISC). The entity types such as PER and MISC are defined for each domain or task.

In this paper, we consider two kinds of annotated corpus; full annotation corpus and partial annotation corpus. In a full annotation corpus, the labels $\boldsymbol{y}$ are given to all the words $\boldsymbol{x}$ (see Fig. 2(a)). Contrarily, in a partial annotation corpus, only some words are labeled (see Fig. 2(b)). As a fully annotated corpus for NER in a general domain, there is a well-known CoNLL-2003 corpus [6] that consists of news articles annotated with four entity types: person, organization, location, and miscellaneous named entity.

Florian et al. [7] used a combination of multiple machine learning models and achieved a high performance of 88.76% in F1 value with the CoNLL-2003 corpus. McCallum et al. [8] considered named entity recognition as a sequence labeling task and proposed a named entity recognition method that uses conditional random fields (CRF). Recently, Ronan et al. [9] proposed a method that uses convolutional neural networks (CNN) for word sequences. Since then, named entity recognition models that use deep learning have become mainstream. Huang et al. [10] proposed a model that substitutes bidirectional long short-term memory (LSTM) for the CNN encoder of Ronan’s model. Lample et al. [11] modeled character and word-level information using bidirectional LSTM. Ma et al. [12] proposed a model that combines bidirectional LSTM with a CNN and CRF and achieved an F1 value of 91.21% with the CoNLL-2003 corpus. Because this model uses CNN for character-level information, a better performance was achieved without pretreatment of data designed by hand. However, these methods aim to construct a named entity recognition model using a full annotation corpus as training data and cannot use a partial annotation corpus. Because of this, annotators have to label all the words in a sentence [13]. CRF in which a partial annotation corpus can be used through margin likelihood have also been proposed, but the training time is very long. For that reason, applying active learning to these models is impractical because it requires repeated learning [14].

2.2. *Active learning*

Active learning is a method for achieving better accuracy with a smaller annotation effort. In the active learning, a machine learning model analyzes an unlabeled dataset and selects a data instance that seems to be the most informative for the current machine learning model. The true label of the selected instance is annotated by an oracle (typically an annotator) who knows the answer. We refer to the action of asking the oracle as a "query" or "request." The instances labeled by the oracle are added to the training dataset, and the model are trained again by using them. Useful labeled data is collected by repeating this process. While there are several active learning scenarios [2], this study uses pool-based active learning. Pool-based active learning selects a data instance for model training in situations in which a large amount of unlabeled data is collected and stored. In scenarios such as pool-based active learning, queries are made based on the evaluation of the information metrics of each instance. A number of query strategies for computing information metrics have been proposed.

Uncertainty sampling [15] is a method to make an inquiry of label of the data instance that has the most uncertain prediction based on the current model. There are various methods of selecting the uncertainty labels. First, the simplest strategy, least confident, is when $\boldsymbol{x}$ is the sentence and $\boldsymbol{y}^*$ is the labels having the highest posterior probability.

$$\phi^{LC}(\boldsymbol{x}) = 1 - P_\theta(\boldsymbol{y}^*|\boldsymbol{x}) \tag{1}$$

When $|\boldsymbol{x}| = |\boldsymbol{y}^*| = 1$, it is regarded as a multi-class classification model. This is the case of the point-wise prediction model.

Another strategy is the margin sampling strategy, which is based on the difference between two labels predicted to be the most probable.

$$\phi^{M}(\boldsymbol{x}) = -\big(P_\theta(\boldsymbol{y}_1^*|\boldsymbol{x}) - P_\theta(\boldsymbol{y}_2^*|\boldsymbol{x})\big) \tag{2}$$

Unlike the least confident strategy, margin sampling accounts for ambiguities other than the most probable label sequence; thus, the amount of information that is taken into account is larger than the least confident strategy.

Another query strategy is query by committee (QBC) [16], which uses multiple models of $\mathcal{C} = \{\theta^{(1)}, \ldots, \theta^{(C)}\}$. The commission that is composed of models predicts the label for each instance and considers the instances in which many different prediction labels have the most useful information. QBC also suggests metrics such as vote entropy [16] and the Kullback-Leibler divergence [17] to assess the degree of discrepancy. There are some other strategies that have been proposed; expected model change [18] is a strategy to query about the data instance that is expected to make the largest impact on the parameter update, and expected error reduction [19] is a strategy to query about the instance that reduces generalization errors.

2.3. *Efficient training of NER models*

The creation of training data for NER models requires expensive cost because it requires the expert knowledge and annotation. Therefore, various methods have been proposed to train NER models efficiently. Patra et al. [20] developed a NER model that can be trained using only information on whether a sentence contains named entities in order to reduce the effort for annotating the position of phrases that indicate named entities. Nguyen et al. [21] used crowdsourcing to develop a NER model that aggregates annotations from multiple workers and allows for training without expert annotator. Lison et al. [22] developed a model that can aggregate predictions provided by multiple NER models trained in different domains, so that predictions can be made with high performance even when there are no training data in the domain to be annotated. A number of studies have also been conducted to apply active learning to NER.

Settles et al. [3] proposed a method of applying active learning to conditional random fields to solve the sequence labeling task. Yanyao et al. [4] have shown that training CNN is faster than training LSTM in deep learning models when multiple named entities need to be extracted. They proposed a CNN-CNN-LSTM model that combines a character CNN, a word CNN, and an LSTM. However, since this model cannot use a partial annotation corpus for training, annotators have to label all the words in a sentence. Therefore, annotators need to label words that are ineffective for training the model, which increases the cost. We in this paper aims to reduce annotation costs in the partial annotation scheme by applying active learning to point-wise prediction model that can be trained by using a partial annotation corpus.

3. Method

3.1. *Named entity recognition with point-wise prediction*

In this paper, we extend the point-wise prediction method proposed by Neubig et al. [23] to extract named entities. This section describes point-wise prediction models. Let $\boldsymbol{x} = \langle x_1, x_2, \ldots, x_T \rangle$ be a sentence and $\boldsymbol{y} = \langle y_1, y_2, \ldots, y_T \rangle$ be a corresponding label (tag) sequence. In point-wise prediction models, the likelihood of the sequence of labels is not considered like in conditional random fields; instead, point-wise prediction models regard this task as a multi-class classification problem for each y_i. Machine learning models such as logistic regression models, support vector machines, and decision trees were developed to deal with multi-class classification.

In this study, we adopt a multi-class logistic regression model as the model for point-wise prediction. Multi-class logistic regression model estimates a single label $y \in \mathcal{Y}$ where $\mathcal{Y} = \{1, 2, \ldots, K\}$. In multi-class logistic regression models, we use the following equation to determine the probability that the class label is y_i when the input is $\boldsymbol{x}$.

$$P(y_i|\boldsymbol{x}) = \frac{\exp\left(\boldsymbol{w}_{y_i}^T \boldsymbol{f}(\boldsymbol{x})\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\boldsymbol{w}_{y'}^T \boldsymbol{f}(\boldsymbol{x})\right)} \tag{3}$$

$\boldsymbol{f}(\boldsymbol{x})$ is a feature vector of which kth element is the value of the kth feature function $f_k(\boldsymbol{x})$ that is defined later. $\boldsymbol{w}$ is a weight vector that corresponds to the feature vector; the weight

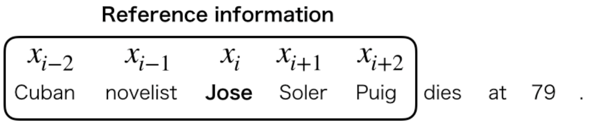


Fig. 3. Information to look up when estimating the label of x_i ($m = 2$).

vector is trained to fit the training data. When training data $(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})$ where $(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(j)}) = (\langle x_1^{(j)}, x_2^{(j)}, \ldots, x_{T_j}^{(j)} \rangle, \langle y_1^{(j)}, y_2^{(j)}, \ldots, y_{T_j}^{(j)} \rangle)$ is given, the weight vector $\boldsymbol{w}$ is optimized by maximizing the log-likelihood function $l(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{y})$. The log-likelihood can be defined as the logarithm of the product of $P(y_i^{(j)}|\boldsymbol{x}^{(j)})$ for all sentences j and positions i in the sentence as below:

$$l(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{y}) = \log \prod_{j=1}^{n} \prod_{i=1}^{T_j} P(y_i^{(j)}|\boldsymbol{x}^{(j)}; \boldsymbol{w}) = \sum_{j=1}^{n} \sum_{i=1}^{T_j} \log P(y_i^{(j)}|\boldsymbol{x}^{(j)}; \boldsymbol{w}) \tag{4}$$

Since the likelihood is a convex function, the weight vector $\boldsymbol{w}$ can be calculated efficiently by using the gradient descent method.

For extracting features on point-wise prediction, we define window width m. The label y_i at position i in a sentence is predicted by using features extracted from the surrounding words $x_{i-m}, \ldots, x_{i+m}$ (see Fig. 3). The following equation is obtained by rewriting multi-class logistic regression in our method.

$$P(y_i|\boldsymbol{x}; \boldsymbol{w}) = P(y_i|x_{i-m}, \ldots, x_{i+m}; \boldsymbol{w}) = \frac{\exp\left(\boldsymbol{w}_y^T \boldsymbol{f}(x_{i-m}, \ldots, x_{i+m})\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\boldsymbol{w}_{y'}^T \boldsymbol{f}(x_{i-m}, \ldots, x_{i+m})\right)} \tag{5}$$

We define the feature functions $\boldsymbol{f} = f_1, \ldots, f_K$ by using "feature templates." Neubig et al. [23] defined feature templates for the point-wise prediction in the domain of Japanese word segmentation. We modify them for the domain of English NER. The feature templates defined in the proposed method are as follows.

- Surface:
  Features based on one-hot encoding of the pairs of (position, vocabulary) that appear in $x_{i-m}, \ldots, x_i, \ldots, x_{i+m}$.
- Word type:
  Information about the word type is extracted from $x_{i-m}, \ldots, x_i, \ldots, x_{i+m}$ and used as the feature. In this paper, we use whether the word begins with a capital letter or consists of only uppercase letters as a feature obtained from the word type.
- Part of speech:
  Features based on one-hot encoding of the pairs of (position, part of speech) that appear in $x_{i-m}, \ldots, x_i, \ldots, x_{i+m}$.

In addition, in the Mori et al. [5] method, a classifier is created for each candidate part of speech of each word, and the part of speech is estimated using the one-to-many method (one-versus-rest). However, our method extracts named entities by using one classifier. We do

Algorithm 1 Active learning by point-wise prediction

```
Require: Labeled data $L$, unlabeled data $U$, query strategy $\phi(\cdot)$, batch size $B$
  repeat
    $\theta \Leftarrow train(L)$
    for $b = 1$ to $B$ do
      $x_b^* \Leftarrow \arg\max_{x \in U} \phi(x)$
      $L \Leftarrow L \cup \{\langle x_b^*, label(x_b^*) \rangle\}$
      $U \Leftarrow U - \{x_b^*\}$
    end for
  until
```

not create a classifier for each label because of the method of labeling unknown words. Mori et al. [5] used point-wise prediction for morphological analysis tasks. Nouns made several appearances in the morphological analysis tasks. Therefore, words that did not appear in the learning corpus or dictionary could be regarded as nouns. On the other hand, in named entity recognition, the preliminary experiment proved that the extraction performance remarkably decreased when the ”O” tag and another specific tag were given to the unknown word. Thus, we use one multivalued classifier for all words.

3.2. *Application to active learning*

Point-wise prediction can be trained by using a partial annotation corpus, so that we can adopt the partial annotation scheme when we apply active learning with the point-wise prediction model. The active learning used in this study utilizes pool-based active learning. Pool-based active learning is an active learning method that assumes there is a small amount of labeled data L and a large amount of unlabeled data pool U. In the pool-based method (based on the prediction of the current model), the data instance that seems to be the most useful for training is selected from the pool and the label is queried of the annotators. The annotators label the instance, and the model is updated accordingly. By repeating these steps, the model is trained. “Sequential active learning” is a strategy that updates the model for each time we obtain a new labeled instance. This method forces the annotator to wait while the model is learning. To reduce the waiting time, the proposed method adopts “batch active learning” that selects and queries multiple unlabeled data instances at a single step.

Pool-based active learning is generally represented by Algorithm 1. First, the algorithm trains the model using the function $train(\cdot)$ for the labeled data L. Next, label queries are made to the annotator B times from the pool U based on the query strategy $\phi(\cdot)$ described in Section 2. The queried data instance is added to L as labeled data and removed from U. The model is repeatedly trained in this loop. Algorithm 1 is regarded as batch active learning when $B > 1$ and sequential active learning when $B = 1$.

Existing models such as CRF and LSTM-CNN-CRF cannot be trained with partially annotated training sentences unless the entire sentence is tagged. Therefore, when active learning is applied, the labels for the all words included in the sentence have to be queried as shown in Fig. 4. On the other hand, when active learning is applied to point-wise prediction, some of the words in the sentence can be separately inquired, as shown in Fig. 5. Active

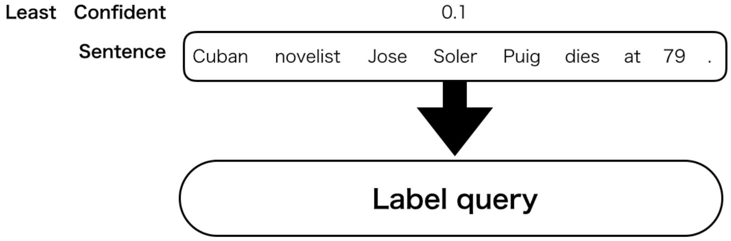


Fig. 4. Label queries in existing models. The existing model calculates the least confident sentence and selects the sentence to be annotated based on it.

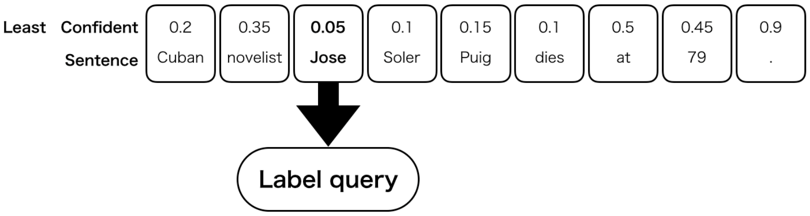


Fig. 5. Label query in point-wise prediction. Since the point-wise prediction can be used to compute the word-by-word least confidence, we can determine the targets to be annotated on a word-by-word basis.

learning with the point-wise prediction models can query more important words than the existing methods when the same number of words are annotated this way. Consequently, the proposed method is expected to be trained more efficiently.

Here, we explain the annotation scheme in the proposed method. Annotators cannot determine whether the ”B” tag or the ”I” tag should be attached unless they check the entire phrase. Thus, we consider a method in which the range of the whole named entity is annotated by showing several surrounding words. The procedure of this “word-by-word query” is as follows: First, the word x_i is selected based on the query strategy. Next, if x_i is in the middle of a named entity, an instance of x_{i-1} is queried of the annotator so that the entire named entity can be annotated. This query is continued until the beginning of the named entity can be seen. If x_i is at the beginning of the named entity or in the middle of the named entity, this function queries x_{i+1}, the next instance in the sequence, until the end of the named entity appears (see Fig. 6).

4. Experiment

4.1. *Dataset*

In this study, we used the CoNLL-2003 English [6] named entity dataset. CoNLL-2003 includes a training set and two test sets. According to the regulation of CoNLL-2003 competition, each set was divided into training data, validation data, and test data. We divided the training data for training the model into two parts: a small amount of labeled data L and a large amount of unlabeled data pool U. Initial training uses only a small amount of labeled data. The validation dataset was used to search the hyperparameters of the deep learning

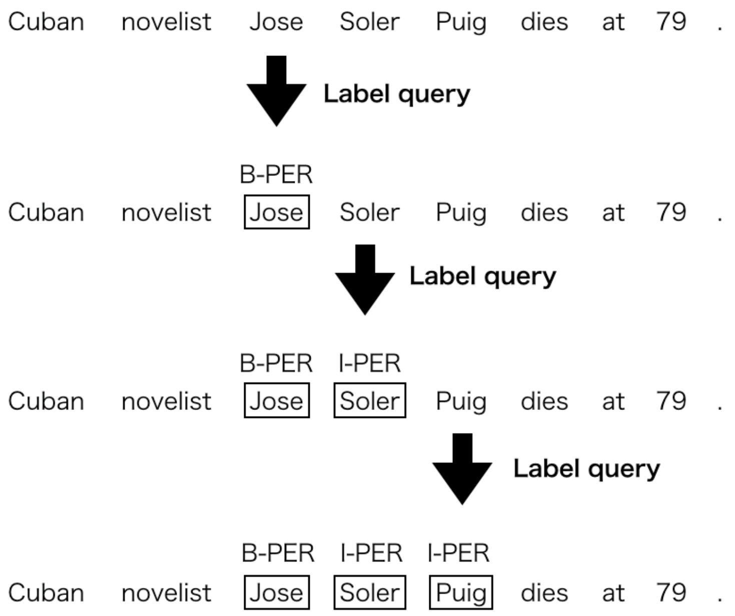

Cuban novelist Jose Soler Puig dies at 79 .
Label query
B-PER
Cuban novelist Jose Soler Puig dies at 79 .
Label query
B-PER I-PER
Cuban novelist Jose Soler Puig dies at 79 .
Label query
B-PER I-PER I-PER
Cuban novelist Jose Soler Puig dies at 79 .

Fig. 6. Example of the label query process. Label queries with point-wise prediction annotate over the selected words.

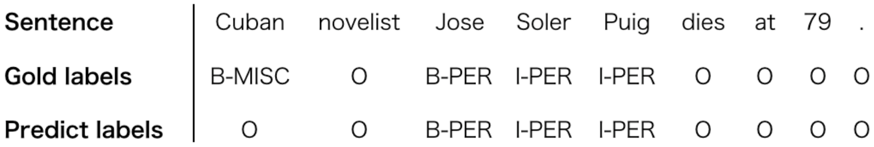
model. In the evaluation, we measured the performance of the named entity recognition for the test data.

4.2. *Procedure of experiment*

We conducted two experiments to examine the effectiveness of the point-wise prediction models with active learning in the partial annotation scheme. In the first experiment, we verified the effectiveness of active learning for NER by point-wise prediction. We compared the random query strategy with the least confident query strategy for point-wise prediction.

In the second experiment, we compared the proposed method with existing methods. We selected the conditional random field and the LSTM-CNN-CRF named entity recognition model proposed by Ma et al. [12] as existing methods. We compared the performance of these methods with the proposed method, the point-wise prediction model, by F1 values. Regarding the hyperparameters of LSTM-CNN-CRF, we set the mini-batch size for training to 10 and the number of epochs to 50.

We chose uncertain sampling as the query strategy. The point-wise prediction model and the conditional random field use the least confidence described in Eq. (1). For the LSTM-CNN-CRF model (for which it is difficult to calculate the probability of the whole sequence), we used margin sampling [24] described in Eq. (2). In all methods, we set up batch-type active learning that queries about 5,000 words at once. In the point-wise prediction, 5,000 words were queried based on the query strategy from all the words in the unlabeled data. On the other hand, the conditional random field and the LSTM-CNN-CRF model could not label words individually. Therefore, labels were added for each sentence, and the batch was



Sentence	Cuban	novelist	Jose	Soler	Puig	dies	at	79	.
Gold labels	B-MISC	O	B-PER	I-PER	I-PER	O	O	O	O
Predict labels	O	O	B-PER	I-PER	I-PER	O	O	O	O

Fig. 7. Example for evalation. In this case, the accuracy is 100%, the recall is 50%, and the F1 value is 66.7%

terminated when the number of labeled words exceeded 5,000.

We performed active learning under simulated conditions using a corpus whose correct answer was known without manual annotation. Therefore, the correct label was always given to words that were queried. In the evaluation, we define a correct extraction of named entity as an extracted phrase that perfectly matches with the start position and the end position of the named entity specified in the ground truth (i.e., CoNLL-2003 dataset). The number of “O” tags that match to the ground truth are not counted as the correct extraction. For example, in Fig. 7, the extraction of the named entity ”Jose Soler Pulg” was correct because it matched ”PER.” Since the model predicted “O” for the word “Cuban” that has “B-MISC” in the ground truth, the extraction for this named entity failed. Therefore, the accuracy is 100%, the recall is 50%, and the F1 value is 66.7%.

5. Result

5.1. *Applying active learning for point-wise predictions*

We compared the point-wise prediction models with and without active learning. In the point-wise prediction model without active learning, the words are selected and queried in random order. Fig. 8 shows the relationship between the extraction performance and the number of words annotated by the simulated annotator. The model using active learning obtained the maximum possible extraction performance with only 10% of the training data pool. Therefore, applying active learning to point-wise prediction reduces the amount of labeled data required, i.e., the burden on the annotator.

5.2. *Performance comparison of named entity recognition using active learning*

Fig. 9 shows the relationship between F1 values and the number of annotated words when active learning was applied to the existing methods and the proposed method. The proposed method had a higher extraction performance than the existing method when there was a small amount of labeled data.

Other methods outperformed the proposed method when the number of annotated words exceeded about 40% of the training data pool. This is the limitation of the proposed method that is caused by the simple architecture of the point-wise prediction model compared with the sentence-wide NER models. We used the margin sampling only for the LSTM-CNN-CRF model. If we apply the margin sampling to point-wise prediction, the performance may be further improved.

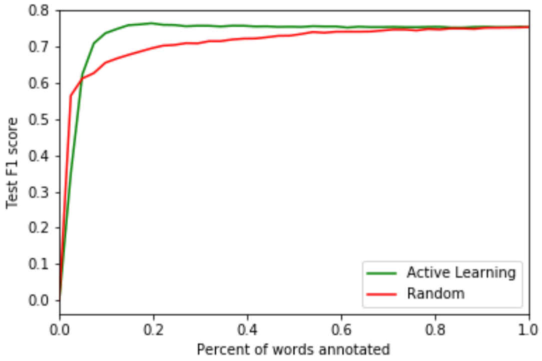


Fig. 8. Experimental results of the query strategy for point-wise prediction.

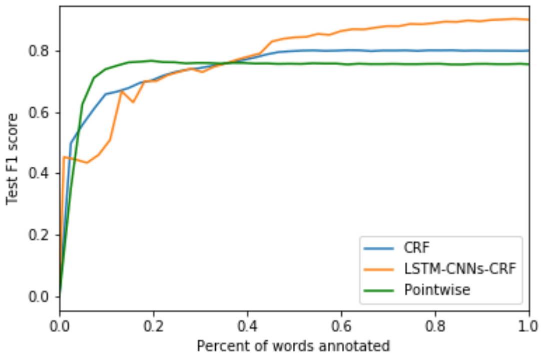


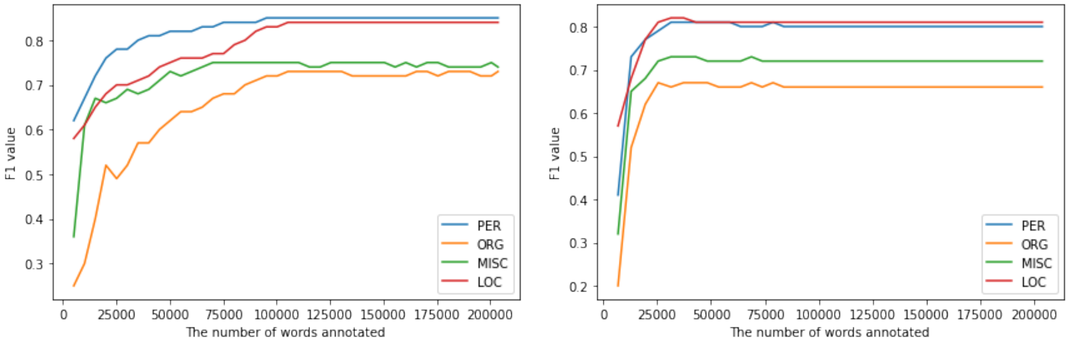
Fig. 9. Experimental results of the existing method and the proposed method.

5.2.1. *Detailed performance evaluation of the proposed method and CRF*

This chapter then provides a more detailed comparative study of the proposed method with the CRF model that is representative of the sentence-wide sequence labeling methods. First, we examine the F1 scores for the named entities labels of each model.

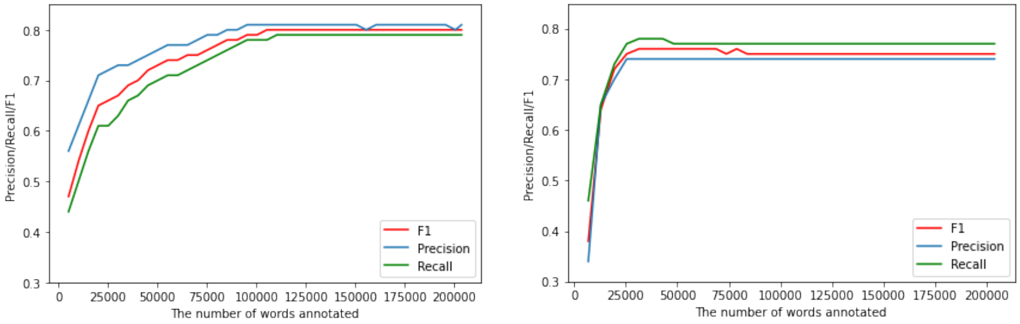
The CoNLL-2003 dataset contains four named entity types: ”PER” for the name of a person, ”LOC” for the name of a place, ”ORG” for the name of an organization, and ”MISC” that is a tag assigned to the other named entities. Fig. 10(a) and 10(b) show the relationship between the number of words annotated and the F1 score for each named entity type by the CRF and the proposed method, respectively. Comparing these two figures, there is no significant difference in the prediction performance of PER, ORG and MISC tags between the proposed method and CRF. However, the improvement in performance of the proposed method for LOC tags is much faster than that of CRF. This is probably because the LOC tags tend to start with a capital letter and the same named entity appears more frequently than other tags.

Next, we examine the detailed evaluation metrics such as recall, precision and F1 score for each model. Fig. 11(a) and 11(b) show the relationship between the number of annotated



(a) CRF method (b) Proposed method

Fig. 10. F1 values of each tag



(a) CRF method (b) Proposed method

Fig. 11. Detailed performance

words and the recall, precision and F1 scores of the CRF and the proposed method. In the result of the CRF, the recall is consistently greater than the precision regardless of the number of annotated words. The difference between precision and recall was large at first, but as the learning converged, the difference becames smaller. In contrast, for the proposed method, the precision is greater than the recall. The difference between precision and recall was almost the same for the proposed method, but it became larger as the learning progressed. This is because the proposed method does not take into account the transition of the predicted tag results, thus the model is not affected by the very frequent tag transition that of "O" to "O" tag transition, i.e., it does not use the property that words that are not named entity are likely to continue.

6. Conclusion

We proposed a new active learning method on the partial annotation scheme that asks human annotators to label a specific part of sentences. The proposed method adopts the point-wise prediction model that can be trained by using partially annotated sentences. The experimental results show that the proposed method can be trained more quickly than the existing active learning methods based on the full annotation scheme. This property is useful when we have a limited budget to develop training data and need to train a custom NER model quickly on the target domain.

The limitation of the proposed method is the upper limit of the performance due to the simple architecture of the point-wise prediction model. When we obtained a sufficient amount of labeled sentences, the NER models based on neural networks outperform the point-wise prediction model. A subject we should address in future work is to overcome this limitation. One potential idea is to switch from the proposed active learning method to neural network models by bridging the gap of the annotation data structure. Another solution is to develop a neural network model that can be trained by using a partial annotation corpus.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 19K20333 and 16H02904, and JST CREST Grant Number JPMJCR16E3 including AIP challenge program, Japan.

1. Diego Mollá, Menno van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, 2006.
2. Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
3. Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, 2008.
4. Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *CoRR*, abs/1707.05928, 2017.
5. Shinsuke Mori, Yosuke Nakata, Graham Neubig, and Tatsuya Kawahara. Morphological analysis with pointwise predictors. *Journal of Natural Language Processing*, 18(4):367–381, 2011.
6. Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, pages 142–147, 2003.
7. Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 168–171, 2003.
8. Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, pages 188–191, 2003.
9. Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.
10. Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.
11. Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, 2016.
12. Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, 2016.
13. Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan 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 (EMNLP)*, pages 864–874, 2014.

14. Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. In *EMNLP*, 2018.
15. David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994.
16. H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 287–294, 1992.
17. Andrew McCallum and Kamal Nigam. Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 350–358, 1998.
18. Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.
19. Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001.
20. Barun Patra and Joel Ruben Antony Moniz. Weakly supervised attention networks for entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6268–6273, 2019.
21. An Thanh Nguyen, Byron Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. Aggregating and predicting sequence labels from crowd annotations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 299–309, 2017.
22. Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. Named entity recognition without labelled data: A weak supervision approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1518–1533. Association for Computational Linguistics, 2020.
23. Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 529–533, 2011.
24. Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318, 2001.