Tag Confidence Measure for Semi-Automatically Updating Named Entity Recognition

Kuniko Saito and **Kenji Imamura**
NTT Cyber Space Laboratories, NTT Corporation
1-1 Hikarinooka, Yokosuka-shi, Kanagawa, 239-0847, Japan
{saito.kuniko, imamura.kenji}@lab.ntt.co.jp

Abstract

We present two techniques to reduce machine learning cost, *i.e.*, cost of manually annotating unlabeled data, for adapting existing CRF-based named entity recognition (NER) systems to new texts or domains. We introduce the *tag* posterior probability as the tag confidence measure of an individual NE tag determined by the base model. Dubious tags are automatically detected as recognition errors, and regarded as targets of manual correction. Compared to entire *sentence* posterior probability, tag posterior probability has the advantage of minimizing system cost by focusing on those parts of the sentence that require manual correction. Using the tag confidence measure, the first technique, known as active learning, asks the editor to assign correct NE tags only to those parts that the base model could not assign tags confidently. Active learning reduces the learning cost by 66%, compared to the conventional method. As the second technique, we propose bootstrapping NER, which semi-automatically corrects dubious tags and updates its model.

1 Introduction

Machine learning, especially supervised learning, has achieved great success in many natural language tasks, such as part-of-speech (POS) tagging, named entity recognition (NER), and parsing. This approach automatically encodes linguistic knowledge as statistical parameters (models) from large annotated corpora. In the NER task, which is the focus of this paper, sequential tagging[1] based on statistical models is similarly used; studies include Conditional Random Fields (CRFs; Lafferty et al., 2001, Suzuki et al., 2006). However, the manual costs incurred in creating annotated corpora are extremely high.

On the other hand, Consumer Generated Media (CGM) such as blog texts has attracted a lot of attention recently as an informative resource for information retrieval and information extraction tasks. CGM has two distinctive features; enormous quantities of new texts are generated day after day, and new vocabularies and topics come and go rapidly. The most effective approach to keep up with new linguistic phenomena is creating new annotated corpora for model re-training at short intervals. However, it is difficult to build new corpora expeditiously because of the high manual costs imposed by traditional schemes.

To reduce the manual labor and costs, various learning methods, such as active learning (Shen et al., 2004, Laws and Schütze, 2008), semi-supervised learning (Suzuki and Isozaki, 2008) and bootstrapping (Etzioni, 2005) have been proposed. Active learning automatically selects effective texts to be annotated from huge raw-text corpora. The correct answers are then manually annotated, and the model is re-trained. In active learning, one major issue is data selection, namely, determining which sample data is most effective. The data units used in conventional methods are sentences.

Automatically creating annotated corpora would dramatically decrease the manual costs. In fact, there always are some recognition errors in any automatically annotated corpus and the editor has to correct errors one by one. Since sentences are used as data units, the editor has to pay attention to all tags in the selected sentence because it is not obvious where the recognition error is. However, it is a waste of manual effort to

[1]Tags are assigned to each input unit (e.g., word) one by one.

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 168–176,
Suntec, Singapore, 7 August 2009.

annotate all tags because most tags must be labeled correctly by the base model[2].

In this paper, we propose a confidence measure based on tag posterior probability for the NER task. Our method does not use the confidence of a sentence, but instead computes the confidence of the tag assigned to each word. The tag confidence measure allows the sentence to which the base model might assign an incorrect tag to be selected automatically. Active learning becomes more efficient because we correct only those tags that have low confidence (cf. Sec. 4).

We can realize the same effect as active learning if we can automatically correct the selected data based upon our tag confidence measure. Our proposal "Semi-Automatically Updating NER" automatically corrects erroneous data by using a seed NE list generated from other information sources. Semi-Automatically Updating NER easily keeps up with new words because it enables us to update the model simply by providing a new NE list (cf. Sec. 5).

2 Named Entity Recognition Task

The NER task is to recognize entity names such as organizations and people. In this paper, we use 17 NE tags based on the IOB2 scheme (Sang and De Meulder, 1999) combined with eight Japanese NE types defined in the IREX workshop (IREX 1999) as shown in Table 1.

For example, "東京 (Tokyo)/ 都 (City)/ に (in)" is labeled like this:

"東京/B-<LOC> 都/I-<LOC> に/O".

This task is regarded as the sequential tagging problem, *i.e.,* assigning NE tag sequences $T = t_1 \cdots t_n$ to word sequences $W = w_1 \cdots w_n$. Recently, discriminative models such as Conditional Random Fields (CRFs) have been successfully applied to this task (Lafferty et al., 2001). In this paper, we use linear-chain CRFs based on the Minimum Classification Error framework (Suzuki et al., 2006). The posterior probability of a tag sequence is calculated as follows:

$$P(T \mid W) = \frac{1}{Z(W)} \exp\{\sum_{i=1}^{n} (\sum_{a} \lambda_a \cdot f_a(t_i, w_i) + \sum_{b} \lambda_b \cdot f_b(t_{i-1}, t_i))\}, \quad (1)$$

where w_i and t_i are the *i*-th word and its corresponding NE tag, respectively. $f_a(t_i, w_i)$ and $f_b(t_{i-1}, t_i)$ is a feature function[3]. λ_a and λ_b is a parameter to be estimated from the training data. $Z(W)$ is a normalization factor over all candidate paths expressed as follows:

$$Z(W) = \sum_{T} \exp\{\sum_{i=1}^{n} (\sum_{a} \lambda_a \cdot f_a(t_i, w_i) + \sum_{b} \lambda_b \cdot f_b(t_{i-1}, t_i))\}. \quad (2)$$

The best tag sequence that maximizes Formula (1) is located using the Viterbi algorithm.

Table 1. NE Types and Tags.

NE Types	NE Tags	
PERSON	B-<PSN>	I-<PSN>
LOCATION	B-<LOC>	I-<LOC>
ORGANIZATION	B-<ORG>	I-<ORG>
ARTIFACT	B-<ART>	I-<ART>
DATE	B-<DAT>	I-<DAT>
TIME	B-<TIM>	I-<TIM>
MONEY	B-<MNY>	I-<MNY>
PERCENT	B-<PCT>	I-<PCT>
outside an NE	O	

3 Error Detection with Tag Confidence Measure

3.1 Tag Posterior Probability

It is quite natural to consider *sentence* posterior probability as a confidence measure of the estimated tag sequences. We focus on *tag* posterior probability, and regard it as the confidence measure of the decoded tag itself. Our method tries to detect the recognition error of each tag by referring to the tag confidence measure.

Figure 1 overviews the calculation of tag confidence measure. The confidence score of tag $t_{i,j}$, which is a candidate tag for word w_i, is calculated as follows:

$$P(t_{i,j} \mid W) = \sum_{T} P(t_{i,j}, T \mid W), \quad (3)$$

where $\sum_{T} P(t_{i,j}, T \mid W)$ is the summation of all NE tag sequences that pass through $t_{i,j}$. This probability is generally called the marginal probability. $j = 1, \cdots, k$ represents the number of NE tags shown in Table 1(*i.e.*, *k*=17 in this paper).

The tag confidence score of $t_{i,j}$ can be calculated efficiently using forward and backward

[2] A base model is the initial model trained with the initial annotated corpora.

[3] We used n-grams (n=1, 2, 3) of surface forms and parts-of-speech within a five word window and 2-gram combinations of NE tags as the feature set.

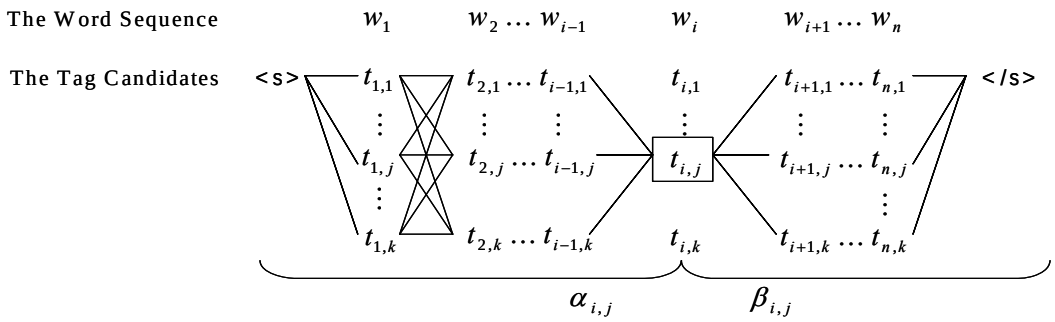

Figure 1. Overview of the tag confidence measure calculation.

algorithms as follows (Manning and Schütze, 1999):

$$P(t_{i,j} \mid W) = \frac{1}{Z(W)} \alpha_{i,j} \cdot \beta_{i,j}, \quad (4)$$

where

$$\alpha_{i,j} = \sum_k \{\alpha_{i-1,k} \cdot \exp\{\sum_a \lambda_a \cdot f_a(t_i, w_i) + \sum_b \lambda_b \cdot f_b(t_{i-1}, t_i)\}\}, \quad (5)$$

$$\beta_{i,j} = \sum_k \{\beta_{i+1,k} \cdot \exp\{\sum_a \lambda_a \cdot f_a(t_{i+1}, w_{i+1}) + \sum_b \lambda_b \cdot f_b(t_i, t_{i+1})\}\}, \quad (6)$$

$$\alpha_{0,j} = 1, \quad (7)$$

$$\beta_{n+1,j} = 1. \quad (8)$$

In this manner, the confidence scores of all tags of each word in a given sentence are calculated. The rejecter then refers to the highest tag confidence score in judging whether the decoded NE tag is correct or incorrect.

3.2 Rejecter

The rejecter tries to detect dubious tags in the NER result derived by the method described in Section 2. For each word, the rejecter refers to the decoded tag t_d, which maximizes Formula (1), and the most confident tag t_1, in terms of the posterior probability as defined in Formula (4). The judgment procedure is as follows:

[1] If t_d is NOT identical to t_1, then t_d is determined to be dubious, and so is rejected as an incorrect tag.[4]
[2] Else, if the confidence score of t_1, called cs_1, is below the predefined threshold, t_d is determined to be dubious, and so is rejected as an incorrect tag.
[3] Otherwise, t_d is accepted as a correct tag.

[4] The decoded tag t_d rarely disagrees with the most confident tag t_1 due to a characteristic of the CRFs.

Increasing the threshold also increases the number of rejected tags and manual annotation cost. In practice, the threshold should be empirically set to achieve the lowest judgment error rate using development data. There are two types of judgment errors: false acceptance and false rejection. False rejection is to reject a correct tag, and false acceptance is to accept an incorrect tag in error. The judgment error rate is taken as the ratio of these two types of errors in all instances.

4 Active Learning

Tag-wise recognition error detection is also helpful for data selection in active learning. If a sentence contains several rejected tags, it contains some new information which the base model does not have. In other words, this sentence is worth learning. Our approach, then, is to base data selection (sentence selection) on the presence of rejected tags. However, it is not necessary to check and correct all tags in each selected sentence. We only have to check and correct the rejected tags to acquire the annotated sentences.

Figure 2 shows our active learning scheme.

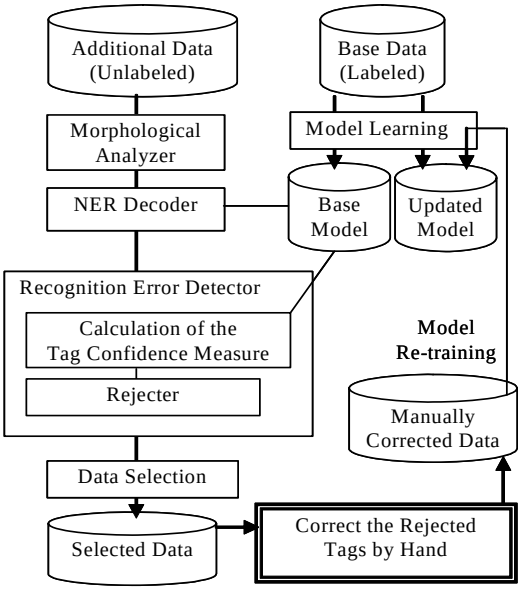

Figure 2. Active Learning Scheme

First, the NER decoder assigns an NE tag to each word[5] of the additional data using the base model trained with the base data. The recognition error detector then determines whether each tag can be confidently accepted as described in Section 3. In this step, the confidence score is calculated using the same base model used for NER decoding. Next, the sentences with at least one rejected tag are selected. Only the rejected tags are manually checked and corrected. Finally, the model is re-trained and updated with the merged data consisting of the manually corrected data and the base data.

4.1 Experiments

We evaluated the efficiency of our active learning method from the perspective of learning cost. A blog corpus consisting of 45,694 sentences in blog articles on the WWW was prepared for the experiments. This corpus was divided into four segments as shown in Table 2. All sentences were manually annotated including additional data. For additional data, these tags were initially hidden and used only for simulating manual correction as shown below. Development data was used for optimizing the threshold by measuring the rejecter's judgment error rate as described in Subsection 3.2.

Table 2. Data Used for Active Learning.

Base Data	11,553 sentences, 162,227 words
Development Data	1,000 sentences, 19,710 words
Additional Data	32,163 sentences, 584,077 words
Test Data	978 sentences, 17,762 words

We estimated the learning cost from the rate of hand-labeled tags. The Word Check Rate (WCR) represents the ratio of the number of the words in the additional data that need to be manually checked and annotated, to the total number of words in the additional data, and is expressed as follows:

WCR= Checked Tags / Total Words.

The system obtained various sizes of selected data as the rejecter changed its threshold from 0.1 to 1.0 for data selection. Only the rejected tags in the selected data were replaced with the tags originally assigned by hand (*i.e.*, correct tags). This procedure simulates manual correction. The manually corrected data was merged with the base data to update the base model.

[5] The morphological analyzer segments an input sentence into a word sequence and assigns parts-of-speech to each word.

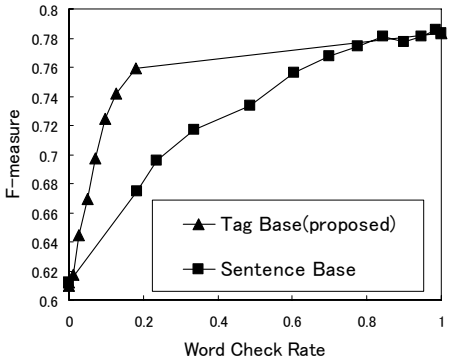

Figure 3. Learning Curves.

We compared our method with data selection based on the sentence confidence measure. Posterior probabilities of sentences were used as the confidence measure, and low-confidence scoring sentences were selected. In contrast to our active learning method, all tags in the selected sentences were replaced with the correct tags in this case.

We evaluated the effectiveness of the updated models against the test data by F-measure as follows:

$$F = \frac{2 \times recall \times precision}{recall + precision}. \quad (9)$$

4.2 Results and Discussions

4.2.1 Learning Curves and Accuracies

Figure 3 shows learning curves of two active learning methods; one is based on our tag confidence measure (Tag Based selection), and the other is based on the sentence confidence measure (Sentence Based selection). In order to reach the F-measure of approximately 0.76, Sentence Based selection requires approximately 60% of the entire data set to be checked by hand. In contrast, Tag Based selection requires only 20% or thereabouts. In other words, our Tag Based selection technique basically matches the performance of Sentence Based selection with only 1/3 of the learning cost.

4.2.2 Types of Tag Replacement

We further investigated the effects of tag-based judgment from the results of an experiment on our Tag Based selection. We categorized tag replacements of the rejected tags into the following four types:

- **No Change**: the rejected tag is replaced with the same tag.
- **O-to-BI**: the rejected tag is an O-tag. It is replaced with a B-tag or an I-tag.

- **BI-to-O**: the rejected tag is a B-tag or an I-tag. It is replaced with an O-tag.
- **BI-to-BI**: the rejected tag is a B-tag or an I-tag. It is replaced with another B-tag or I-tag.

Table 3 shows the distribution of these four categories in the selected data for the threshold of 0.5. This threshold achieves the lowest judgment error rate given the development set.

The rate of No Change replacement type is the highest. This means that the rejecter rejected too many tags, which actually did not need to be checked by hand. Although this result does not have a negative influence on the accuracy of the updated model, it is not preferable from the learning cost perspective. Further consideration should be given in order to improve the rejecter's judgment.

O-to-BI type accounts for the 2nd highest percentage of all replacements: it is almost one third of all changes. Excluding No Change type (*i.e.,* among O-to-BI, BI-to-O and BI-to-BI types), O-to-BI type makes up nearly 60% of these three replacement types. This result shows that there were many new NEs not recognized by the base model in the selected data.

Table 3. The Distribution of Replacement Types.

Replacement Type	Frequency	%
No Change	13,253	43.6
O-to-BI	10,042	33.0
BI-to-O	2,419	8.0
BI-to-BI	4,688	15.4
Total	30,402	100.0

5 Bootstrapping for NER

As mentioned in Section 4, we have to correct an O-tag to a B-tag or an I-tag in many cases, almost 60% of all actual corrections. This situation arises from a characteristic of the NER task. In the NER task, most NE tags in the entire corpus are O-tags. In fact, we found that 91 % of all tags were O-tags in the additional data discussed in Section 4. Thus, when a new NE appears in a sentence, this new NE is often mistakenly given an O-tag by the base model.

The fact that only O-tags are dominant implies that we have a chance to find a correct B-tag or I-tag when we look up the 2nd candidate. This is because one of these top two candidates is inevitably a B-tag or an I-tag. Thus, it is valuable to consider what the NEXT preferable tag is when the most preferable tag is rejected.

We examined in detail the accuracy of the tag candidates when the threshold is 0.5 as summarized in Table 4. When the top tag (*i.e.,* the tag with the highest tag confidence score) is accepted, its accuracy is 94 %, obviously high. On the other hand, the top tag's accuracy is only 43 % when it is rejected. However, focusing both on the top tag and on the 2nd tag provides an opportunity to correct the rejected tag in this case. If we consider these top two tags together when the 1st tag is rejected, the possibility of finding the correct tag is 72 %, relatively high. This suggests that the system is capable of correcting the rejected tag automatically by using the top two tag candidates. On this background, automatic correction is attempted for re-training the model through the use of a bootstrapping scheme.

Table 4. Accuracy of the Tags.

Rejecter's Judgment of the Top Tag		
ACCEPT	REJECT	
Top Tag	Top Tag	2nd Tag
94 %	43 %	29 %

Figure 4 shows an example of the top two tag candidates and their tag confidence scores when the top tag's confidence score is lower than the threshold (=0.5). We call this lattice the "tag graph" in this paper. The system failed to recognize the movie title "3丁目の夕日" ("*Sanchome no Yuuhi*", which means "Sunset on Third Street") as ARTIFACT only with the top tag candidates. However, it may find a correct tag sequence using the top two tag candidates (shaded cells in Figure 4). Once the system identifies the correct tag sequence automatically in the tag graph, the sequence is used as a manually annotated sequence. We introduce this new technique, Semi-Automatically Updating NER.

Figure 4. The Top Two Tag Candidates with Tag Confidence Measures.

	Top Tag		2nd Tag	
	Tag	score	Tag	score
今日(Today)	B-<DAT>	0.95		
「(")	O	0.98		
3(Third)	O	**0.47**	**B-<ART>**	0.36
丁目(Street)	O	**0.38**	**I-<ART>**	0.36
の(on)	O	**0.49**	**I-<ART>**	0.38
夕日(Sunset)	**I-<ART>**	**0.39**	O	0.34
」(")	O	0.99		
が(is)	O	0.99		
放映(broadcast)	O	0.99		

5.1 Semi-Automatically Updating NER

By extracting the correct tag sequence in each tag graph as shown in Figure 4, it is possible to obtain automatically corrected data, which also serve as new training data. Based on this idea, we propose Semi-Automatically Updating NER, which is hereafter simply referred to as Updating NER.

Figure 5 overviews Updating NER. The rejecter produces the sentences with tag graphs based on the tag confidence measure. In this new procedure, however, the rejecter's role differs from that described in Section 4 as follows:

[1] When the highest confidence score cs_1 equals or exceeds the threshold, the rejecter accepts only the top candidate tag t_1, otherwise it goes to Step 2.
[2] When cs_1 is less than the threshold, the rejecter accepts not only the top tag t_1 but also the 2nd tag t_2.

Sentences that contain the 2nd candidates are selected in data selection for subsequent processing. The correct tag sequence in each tag graph is identified in automatic correction as follows:

[1] Select the tag sequence that has the longest[6] and consistent NE from the tag graph.
[2] If the longest NE also exists in a seed NE list, which will be described below, the system extracts the entire sentence with its tag sequence as corrected data.

In Step 1, the system selects one preferable tag sequence based on the longest NE match. In the tag graph shown in Figure 4, there are 16 possible sequences because four words "3", "丁目(Street)", "の(on)" and "夕日(Sunset)" each have two tag candidates; O or B for "3", O or I for "丁目(Street)" and "の(on)", and I or O for "夕日(Sunset)". For example, "B I I I", "B I I O", "B I O O", "O O O I", "O O O O" and the rest. Because the sequence "B I I I" constructs the longest NE, the system selects the tag sequence that contains the ARTIFACT "3丁目の夕日." Other sequences that contain partial NEs such as "3", "3丁目", "3丁目の", which are all ARTIFACTs, are ignored.

In Step 2, the system judges whether the tag sequence selected in Step 1 is indeed correct.

[6] By longest, we mean the longest tag sequence that does *not* include any O-tags.

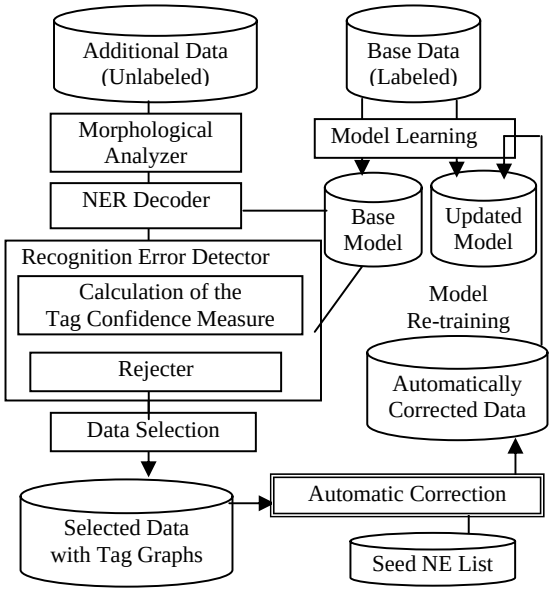

Figure 5. Semi-Automatically Updating NER Scheme.

However, the system requires some hints to judge the correctness, so we need to prepare a seed NE list, which contains surface forms and NE types. This list can be created by manually annotation of possible NEs or automatic generation from other sources such as dictionaries. When the same NE exists both in the selected tag sequence and the seed NE list, the system regards the selected tag sequence as reliable and extracts it as automatically corrected data. Finally, the model is updated by merging the automatically corrected data with the base data.

Bootstrapping means that data selection and correction of the selected data are completely automatic; we still have to prepare the seed NE list somehow. Thus the learning cost is quite low because we only need to provide an NE list as a seed. Updating NER is capable of modifying the model to keep up with the emergence of new named entities. Therefore, it is effective to analyze the large amount of texts that emerge everyday, such as blogs on the WWW.

5.2 Experiments

We tested our Updating NER with a large amount of blog texts from the WWW. One week's worth of blog texts was crawled on the WWW to generate the additional data. Table 5 shows the statistics of the data used in our experiments. The test data contained only the blog texts generated in December 2006, and the base data is about a half year older than the test data. Therefore, it is difficult for the base model to recognize new NEs in the test data. One week's

worth of December 2006 blog texts were prepared for bootstrapping. The overlap between the test data and the additional data was removed in advance. We set the rejecter's threshold at 0.5 and selected the data with tag graphs from the additional data.

Japanese Wikipedia entries were used as the seed NE list. The titles of Wikipedia articles were regarded as surface forms. NE types were estimated from the category sections of each article, based on heuristic rules prepared in advance. We collected 104,296 entries as a seed NE list.

Using this seed list, Updating NER extracted the seed NE and its context from the selected data automatically. If the system found a match, it extracted the sentence with its tag sequence from the selected data. The automatically corrected data was then merged with the base data in order to re-train the base model.

For comparison, we evaluated the effect of the seed NE list itself. If there is a sequence of words that can be found in the seed list, then that sequence is always recognized as a NE. Note that the other words are simply decoded using the base model. We call this method 'user dictionary'. Here, we use recall and precision to evaluate the accuracy of the model.

Table 5. Data Description for Updating NER.

Base Data (blog in Sep. 04-Jun. 06)	43,716 sentences 746,304 words
Additional Data (one week's blog in Dec. 06)	240,474 sentences 3,677,077 words
Selected Data from the Additional Data	113,761 sentences 2,466,464 words
Test Data (blog in Dec.06)	1,609 sentences 21,813 words

5.3 Results

Table 6 shows the details of accuracy results regarding the following four NE types: PERSON, LOCATION, ORGANIZATION, and ARTIFACT, which are referred to hereafter as PSN, LOC, ORG and ART, respectively. Although we added Wikipedia as a user dictionary to the base model, it only slightly improved the recall. In fact, it has no positive and sometimes a negative effect on precision (e.g., ART decreased from 0.666 to 0.619). This indicates that adding an NE list as a dictionary is not enough to improve the accuracy of a NER system. This is because the NER system cannot discriminate an NE from surrounding unrelated words. It simply extracts matched sequences of words, so it overestimates the number of NEs.

On the contrary, our Updating NER improved both recall and precision (e.g., the recall and the precision in ART improved from 0.320 to 0.364 and from 0.666 to 0.694, respectively.). This means that not only the NE list but also the contexts are actually needed to retrain the model. Our Updating NER scheme has the advantage of finding the reliable context of a seed NE list automatically. Although some manual effort is needed to provide a seed NE list, its associated cost is lower than the cost of annotating the entire training data. Thus, we regard Updating NER as a promising solution for reducing learning cost in practical NER systems.

As shown in Table 6, neither user dictionary method nor Updating NER improves the accuracy in ORG. We assume that this is caused by the distribution of NE types in the seed NE list. In the seed list selected from the Wikipedia entries, PSN-type is dominant (74%). ORG-type is scant at only 11%, so the system did not have enough chances to retrain the ORG-type. Rather, it might be the case that the system had a tendency to recognize ORG-type as PSN-type because peoples' names are often used as organization names. Further investigation is needed to clarify the impact of the distribution and the quality of the seed NE list.

Table 6. Details of Accuracy.

		PSN	LOC	ORG	**ART**
Base Model	rec.	0.640	0.737	0.688	0.320
	prec.	0.699	0.811	0.652	0.666
+Wikipedia (user dic.)	rec.	0.686	0.729	0.688	0.354
	prec.	0.716	0.815	0.654	0.619
+Wikipedia (UpdatingNER)	rec.	0.649	0.747	0.678	0.364
	prec.	0.728	0.822	0.632	0.694

5.4 Discussions

Compared to conventional machine learning techniques, the most distinctive feature of Updating NER is that the system can focus on the top two candidates when the confidence score of the top candidate is low. This feature actually has a great advantage in the NER task, because the system is capable of determining what the next preferable tag is when a new NE appears which is assigned an O-tag by the base model.

Updating NER, however, has one weak point. That is, the following two strict conditions are required to correct the selected data automatically. First, the correct tag sequence must appear in tag graphs (*i.e.,* as one of the top two tag candidates). Second, the NE must also appear in the seed NE list. These conditions decrease the

chance of extracting sentences with correct tag sequences from the selected data.

To overcome this weakness, one practical approach is to use Updating NER in combination with active learning. In the case of active learning, we do not need the correct tags in the top two candidates. The editor can assign correct tags without considering the order of candidates. In short, active learning has broad coverage in terms of learning, while Updating NER does not. Therefore, active learning is suitable for improving the performance level of the entire base model. Updating NER has the advantage of staying current with new named entities which emerge every day on the WWW. In practical use, for example, it will be better to update the model every week with Updating NER to keep up with new named entities, and occasionally perform active learning (every six months or so) to enhance the entire model. In the future, we plan to evaluate the efficiency of our two learning methods in practical applications, such as domain adaptation and acquisition of hot trend NE words from blog texts on the WWW.

6 Related Works

To date, there have been many related works on active learning not only for the NER task (Shen et al., 2004, Laws and Schütze, 2008) but also for other tasks, such as POS tagging (Engelson and Dagan, 1996), text classification (Lewis and Catlett, 1994), parsing (Hwa, 2000), and confusion set disambiguation (Banko and Brill, 2001). Active learning aims at effective data selection based on criterion measures, such as the confidence measure. Most previous works focus on the *Sentence*-Based criterion evaluation and data selection. Our proposal differs from those previous works in that we focus on the *Tag*-Based strategy, which judges whether each tag should be accepted or rejected. This approach maximizes the effectiveness of manual annotation by leaving the accepted tags in without any manual correction. As a result, our Tag-based approach reduces the manual annotation cost by 66 %, compared to the Sentence-Base method.

Semi-supervised learning has become an active area in machine learning; it utilizes not only annotated corpora but also huge amounts of plain text for model training. Several studies adapted semi-supervised learning to suit NLP tasks, such as word sense disambiguation (Yarowsky, 1995), text classification (Fujino et al., 2008), and chunking and NER (Suzuki and Isozaki, 2008). Suzuki and Isozaki (2008) suggest that a GIGA-word size plain text corpus may further improve the performance of the state-of-the-art NLP system. In this paper, however, we aim at model adaptation to the CGM domain to keep up with the new linguistic phenomena that are emerging every day. Because it is difficult to obtain GIGA-word size plain text sets that reflect such new linguistic phenomena, it is not practical to directly apply this approach to our task.

Bootstrapping is similar to semi-supervised learning in that it also allows the use of plain text (Etzioni 2005, Pantel and Pennacchioti 2006). In this learning method, it is possible to extract new instances automatically from plain text with small seed data prepared manually. Our Updating NER is similar to bootstrapping in that it extracts new annotated corpora automatically from plain text data starting with a seed NE list. However, the goal of conventional bootstrapping is to develop a new dictionary or thesaurus by extracting new instances. On the contrary, our goal is to acquire a new NE and its surrounding context in a sentence, not to build a NE dictionary (*i.e.*, correct tag sequence). It is the tag sequence and not a single NE that is needed for model training. Updating NER is a novel approach in the point of applying bootstrapping to the framework of supervised learning. This approach is quite effective in that it has the advantage of reducing learning cost compared with active learning because only a seed NE list is needed.

7 Conclusions

To reduce machine learning cost, we introduced two techniques that are based on a tag confidence measure determined from tag posterior probability. Dubious tags are automatically detected as recognition errors using the tag confidence measure. This approach maximizes the effectiveness of manual annotation by leaving the confident tags in without any manual correction.

We first applied this technique to active learning by correcting error tags manually. We found that it matches the performance of the learning method based on the sentence confidence measure with only 1/3 of the learning cost.

Next, we proposed Semi-Automatic Updating NER which has a bootstrap learning scheme, by expanding the scope from the top tag candidate to include the 2nd candidate. With this new scheme, it is possible to collect auto-labeled data from a large data source, such as blog texts on the WWW, by simply providing a seed NE list.

References

M. Banko and E. Brill. 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In *Proc. of ACL-2001*, pages 26-33.

S. A. Engelson and I. Dagan. 1999. Committee-Based Sample Selection for Probabilistic Classifiers. *Journal of Artificial Intelligence Research*, vol.11(1999), pages 335-360.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 165(1), pages 91-134.

A. Fujino, N. Ueda, and K. Saito. 2008. Semisupervised Learning for a Hybrid Generative /Discriminative Classifier Based on the Maximum Entropy Principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(3), pages 424-437.

R. Hwa. 2000. Sample Selection for Statistical Grammer Induction. In *Proc. of EMNLP/VLC-2000*, pages 45-52.

IREX Committee (ed.), 1999. In *Proc. of the IREX workshop*. http://nlp.cs.nyu.edu/irex/

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML-2001*. pages 282-289.

F. Laws and H. Schütze. 2008. Stopping Criteria for Active Learning of Named Entity Recognition. In *Proc. of COLING-2008*, pages 465-472.

D. Lewis and J. Gatlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proc. of ICML-1994*, pages 148-156.

C. D. Manning and H. Schütze. 1999. Foundations of Statistical Natural Language Processing. The MIT Press.

P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proc. of COLING-ACL-2006*, pages 113-120.

E. F. T. K. Sang and F. De Meulder. 1999. Representing text chunks. In *Proc. of EACL-1999*, pages 173-179.

D. Shen, J. Zhang, J. Su, G. Zhou, and C. L. Tan. 2004. Multi-Criteria-based Active Learning for Named Entity Recognition. In *Proc. of ACL-2004*, pages 589-596.

J. Suzuki and H. Izozaki. 2008. Semi-Supervised Sequential Labeling and Segmentation using Gigaword Scale Unlabeled Data. In *Proc. of ACL-2008*, pages 665-673.

J. Suzuki, E. McDermott, and H. Isozaki. 2006. Training Conditional Random Fields with Multivariate Evaluation Measures. In *Proc. of COLING-ACL-2006*. pages 617-624.

D. Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proc. of ACL-1995*, pages 189-196.

X. Zhu. 2007. Semi-Supervised Learning, ICML-2007 Tutorial.