# Supervised and Semi-Supervised Sequence Learning for Recognition of Requisite Part and Effectuation Part in Law Sentences

**Le-Minh Nguyen** and **Ngo Xuan Bach** and **Akira Shimazu**
Japan Advanced Institute of Science and Technology (JAIST)
Asahidai 1-1, Nomi, Ishikawa, 923-1292 Japan
{nguyenml,bachnx,shimazu}@jaist.ac.jp

## Abstract

Analyzing the logical structure of a sentence is important for understanding natural language. In this paper, we present a task of Recognition of Requisite Part and Effectuation Part in Law Sentences, or RRE task for short, which is studied in research on Legal Engineering. The goal of this task is to recognize the structure of a law sentence. We empirically investigate how the RRE task is conducted with respect to various supervised machine learning models. We also compared the impact of unlabeled data to RRE tasks. Experimental results for Japanese legal text domains showed that sequence learning models are suitable for RRE tasks and unlabled data also significantly contribute to the performance of RRE tasks.

## 1 Introduction

Legal Engineering (Katayama 07) is a new research field which aims to achieve a trustworthy electronic society. There are two important goals of Legal Engineering. The first goal is to help experts make complete and consistent laws, and the other is to design an information system which works based on laws. To achieve this we need to develop a system which can process legal texts automatically.

Legal texts have some specific characteristics that make them different from other daily-use documents. Legal texts are usually long and complicated. They are composed by experts who spent a lot of time to write and check them carefully. One of the most important characteristics of legal texts is that law sentences have some specific structures. In most cases, a law sentence can roughly be divided into two parts: a *requisite part* and an *effectuation part* (Nakamura et al., 07; Tanaka et al., 93). For example, the Hiroshima city provision 13-2 *When the mayor designates a district for promoting beautification, s/he must in advance listen to opinions from the organizations and the administrative agencies which are recognized to be concerned with the district*, includes a requisite part (before the comma) and an effectuation part (after the comma) (Nakamura et al., 07).

The requisite part and the effectuation part of a law sentence are composed from three parts: a *topic part*, an *antecedent part*, and a *consequent part*. There are four cases (illustrated in Figure 1) basing on where the topic part depends on: case 0 (no topic part), case 1 (the topic part depends on the antecedent part), case 2 (the topic part depends on the consequent part), and case 3 (the topic part depends on both the antecedent part and the consequent part). In case 0, the requisite part is the antecedent part and the effectuation part is the consequent part. In case 1, the requisite part is composed from the topic part and the antecedent part, while the effectuation part is the consequent part. In case 2, the requisite part is the antecedent part, while the effectuation part is composed from the topic part and the consequent part. In case 3, the requisite part is composed from the topic part and the antecedent part, while the effectuation part is composed from the topic part and the consequent part. Figure 2 gives examples of law sentences in four cases.

Analyzing the logical structure of law sentences is an important task in Legal Engineering. This task is



*Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*, pages 21–29,
Blois (France), July 12-15, 2011. 

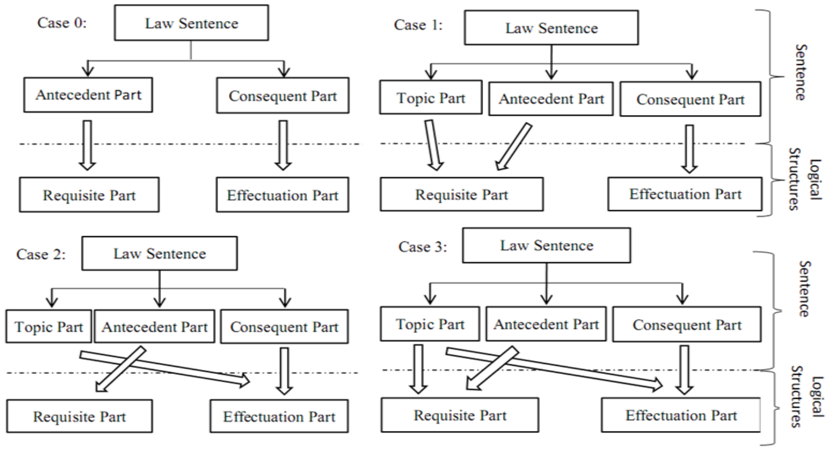


Figure 1: Four cases of the logical structure of a law sentence.

**Case 0:**
<A>被保険者期間を計算する場合には、</A><C>月によるものとする。</C>
<A>When a period of an insured is calculated,</A><C> it is based on a month.</C>

**Case 1:**
<A>被保険者の資格を喪失した後、さらにその資格を取得した</A><T1>者については、</T1><C>前後の被保険者期間を合算する。</C>
<T1>For the person</T1>
<A>who is qualified for the insured after s/he was disqualified, </A>
<C>the terms of the insured are added up together. </C>

**Case 2:**
<T2>この法律による年金の額は、</T2><A>国民の生活水準その他の諸事情に著しい変動が生じた場合には、</A><C>変動後の諸事情に応ずるため、速やかに改定の措置が講ぜられなければならない。</C>
<T2>For the amount of the pension by this law, </T2>
<A>when there is a remarkable change in the living standard of the nation or the other situations, </A>
<C>a revision of the amount of the pension must be taken action promptly to meet the situations. </C>

**Case 3:**
<T3>政府は、</T3><A>第一項の規定により財政の現況及び見通しを作成したときは、</A><C>遅滞なく、これを公表しなければならない。</C>
<T3> For the Government, </T3>
<A>when it makes a present state and a perspective of the finance, </A>
<C>it must announce it officially without delay.</C>

Figure 2: Examples of four cases of the logical structure of a law sentence. *A* means *antecedent part*, *C* means *consequent part*, and *T1*, *T2*, *T3* mean *topic parts* which correspond to case 1, case 2, and case 3 (the translations keep the ordinal sentence structures).

a preliminary step to support tasks in legal text processing, such as translating legal articles into logical and formal representations and verifying legal documents, legal article retrieval, legal text summarization, question answering in legal domains, etc (Katayama 07; Nakamura et al., 07). In a law sentence, the consequent part usually describes a law provision, and the antecedent part describes cases in which the law provision can be applied. The topic part describes the subjects which are related to the law provision. Hence, the outputs of the RRE task will be very helpful to not only lawyers but also people who want to understand the law sentence. They can easily understand 1) what does a law sentence say? 2) what cases in which the law sentence can be applied? and 3) what subjects are related to the provision described in the law sentence?

In this paper, we present a task of *Recognition of Requisite Part and Effectuation Part in Law Sentences* - the RRE task. We show how to model this task by using sequence learning models. The first one applies Conditional Random Fields (CRFs), a special version of conditionally-trained finite state machines. We studies two different machine learning models for CRFs. The second one focus on discriminative sequence learning models using online learning framework (Crammer et al, 2006). We then empirically investigate several sequence learning models for RRE task. In addition, We depict a simple semi-supervised learning method for the RRE task using the Brown clustering algorithm. We also show experimental results on an annotated corpus of Japanese national pension law sentences. Our models achieved 88.58% (using supervised learning) and 88.84% (using semi-supervised learning) in the $F_{\beta=1}$ score.

The remainder of this paper is organized as follows. First, Section 2 presents how to model the RRE task as a sequence labeling problem, and shows experimental results about the effect of features on the task. Next, we describe another setting for the task based on Bunsetsu (chunks in English) in Section 3. Then, Section 5 describes a simple semi-supervised learning method for the task. Finally, some conclusions are given in Section 6.

## 2 RRE as a Sequence Learning Problem

### 2.1 Problem Setting

Let $x$ be an input law sentence in a law sentence space $X$, then $x$ can be represented by a sequence of words $[w_1, w_2, \ldots, w_n]$. Let $P$ be the set of predefined logical part categories. A logical part $p(s, e)$ is the sequence of consecutive words spanning from word $w_s$ to word $w_e$ with category $p \in P$.

We define two kinds of relationships between two logical parts: *overlapping* and *embedded*. Let $p_1(s_1, e_1)$ and $p_2(s_2, e_2)$ be two different logical parts of one sentence $x$. We say that $p_1(s_1, e_1)$ and $p_2(s_2, e_2)$ are *overlapping* if and only if $s_1 < s_2 \leq e_1 < e_2$ or $s_2 < s_1 \leq e_2 < e_1$. We denote the *overlapping* relationship by $\sim$. We also say that $p_1(s_1, e_1)$ is *embedded* in $p_2(s_2, e_2)$ if and only if $s_2 \leq s_1 \leq e_1 \leq e_2$, and denote the *embedded* relationship by $\prec$.

In the RRE task, we try to split a source sentence into some *non-overlapping* and *non-embedded* logical parts. Let $S$ be the set of all possible logical parts, $S = \{p(s, e) | 1 \leq s \leq e \leq n, p \in P\}$. A *solution* of the RRE task is a subset $y \subseteq S$ which does not violate the *overlapping* relationship and the *embedded* relationship. Formally, the *solution space* can be described as follows: $Y = \{y \subseteq S | \forall u, v \in y, u \nsim v, u \nprec v\}$. The learning problem in the RRE task is to learn a function $R : X \to Y$ from a set of $m$ training samples $\{(x^i, y^i) | x^i \in X, y^i \in Y, \forall i = 1, 2, \ldots, m\}$.

One important characteristic of our task is that the input sentences are usually very long and complicated, so the logical parts are also long.

We model the RRE task as a sequence labeling problem, in which each sentence is a sequence of words. Figure 3 illustrates an example in IOB notation. In this notation, the first word of a part is tagged by $B$, the other words of the part are tagged by $I$, and a word not included in any part is tagged by $O$. This law sentence consists of an antecedent part (tag $A$) and a consequent part (tag $C$) (we will use this example for all the sections of this paper).

In the RRE task, we consider two types of law sentences: *implication type*[1] and *equivalence type*. Figure 4 shows the logical structure of a law sentence in the equivalence type. In this type, a sentence consists of a *left equivalent part* and a *right equivalent part*. The requisite part is the left equivalent part, and the effectuation part is the right equivalent part. In all, we have 7 kinds of parts, as follows:

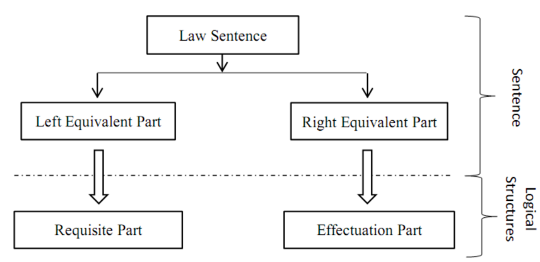


Figure 4: The logical structure of a law sentence in the equivalence type.

1. Implication sentences:
   - Antecedent part ($A$)
   - Consequent part ($C$)
   - Three kinds of topic parts $T_1$, $T_2$, $T_3$ (correspond to case 1, case 2, and case 3)
2. Equivalence sentences:
   - The left equivalent part ($EL$)
   - The right equivalent part ($ER$)

[1]The logical structure of a law sentence in this type is shown in Figure 1.

| Source Sentence | <A>被保険者期間を計算する場合には、</A><C>月によるものとする。</C> (*When a period of an insured is calculated, it is based on a month.*) | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Word Sequence | 被 *hi* | 保険 *hoken* | 者 *sha* | 期間 *kikan* | を *wo* | 計算 *keisan* | する *suru* | 場合 *baai* | に *ni* | は *wa* | 、 | 月 *tsuki* | による *niyoru* | もの *mono* | と *to* | する *suru* | 。 | |
| Tag Sequence | B-A | I-A | I-A | I-A | I-A | I-A | I-A | I-A | I-A | I-A | I-A | B-C | I-C | I-C | I-C | I-C | I-C | |

Figure 3: A law sentence in the IOB notation.

In the IOB notation, we will have 15 kinds of tags: B-A, I-A, B-C, I-C, B-$T_1$, I-$T_1$, B-$T_2$, I-$T_2$, B-$T_3$, I-$T_3$, B-EL, I-EL, B-ER, I-ER, and O[2]. For example, an element with tag B-A begins an antecedent part, while an element with tag B-C begins a consequent part.

### 2.2 Discriminative Sequence Learning Models

In this section, we briefly introduce three discriminative sequence learning models for RRE problems.

#### 2.2.1 Conditional Random Fields

*Conditional Random Fields* (CRFs) (Lafferty et al., 01) are undirected graphical models used to calculate the conditional probability of values on designated output nodes, given values assigned to other designated input nodes for data sequences. CRFs make a first-order Markov independence assumption among output nodes, and thus correspond to finite state machine (FSMs).

Let **o** = $(o_1, o_2, \ldots, o_T)$ be some observed input data sequence, such as a sequence of words in a text (values on $T$ input nodes of the graphical model). Let **S** be a finite set of FSM states, each is associated with a label $l$ such as a clause start position. Let **s** = $(s_1, s_2, \ldots, s_T)$ be some sequences of states (values on T output nodes). CRFs define the conditional probability of a state sequence given an input sequence to be

$$P_\Lambda(s|o) = \frac{1}{Z_o} exp\left(\sum_{t=1}^{T} F(s,o,t)\right) \qquad (1)$$

where $Z_o = \sum_s exp\left(\sum_{t=1}^{T} F(s,o,t)\right)$ is a normalization factor over all state sequences. We denote $\delta$ to be the Kronecker-$\delta$. Let $F(s,o,t)$ be the sum of CRFs features at time position $t$:

$$\sum_i \lambda_i f_i(s_{t-1}, s_t, t) + \sum_j \lambda_j g_j(o, s_t, t) \qquad (2)$$

[2]Tag O is used for an element not included in any part.

where $f_i(s_{t-1}, s_t, t) = \delta(s_{t-1}, l')\delta(s_t, l)$ is a *transition* feature function which represents sequential dependencies by combining the label $l'$ of the previous state $s_{t-1}$ and the label $l$ of the current state $s_t$, such as the previous label $l' = B - A$ and the current label $l = I - A$. $g_j(o, s_t, t) = \delta(s_t, l)x_k(o, t)$ is a *per-state* feature function which combines the label $l$ of current state $s_t$ and a context predicate, i.e., the binary function $x_k(o,t)$ that captures a particular property of the observation sequence o at time position $t$. For instance, the current label is $B - A$ and the current word is *"conditional"*.

Training CRFs is commonly performed by maximizing the likelihood function with respect to the training data using advanced convex optimization techniques like L-BFGS. Recently, several works apply Stochastic Gradient Descent (SGD) for training CRFs models. SGD has been historically associated with back-propagation algorithms in multilayer neural networks.

Inference in CRFs, i.e., searching the most likely output label sequence of an input observation sequence, can be done using the Viterbi algorithm.

#### 2.2.2 Online Passive-Aggressive Learning

Online Passive-Aggressive Learning (PA) was proposed by Crammer (Crammer et al, 2006) as an alternative learning algorithm to the maximize margin algorithm. The PA algorithm has been shown to be successful for many sequence classification tasks. The details of the PA algorithm for RRE task are presented as follows.

Assume that we are given a set of sentences $x_i$ and their labels $y_i$ where $i = 1, ..., n$. Let the feature mapping between a sentence $x$ and a sequence of labels $y$ be: $\Phi(x,y) = \Phi_1(x,y), \Phi_2(x,y), ..., \Phi_d(x,y)$ where each feature mapping $\Phi_j$ maps $(x,y)$ to a real value. We assume that each feature $\Phi(x,y)$ is associated with a weight value. The goal of PA learning for sequence learning tasks is to obtain a parameter $w$ that minimizes the

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Sentence | <A>被保険者期間を計算する場合には、</A><C>月によるものとする。</C> *(When a period of an insured is calculated, it is based on a month.)* | | | | | | | | | | | | | | | | |
| Original Sequence | 被 *hi* | 保険 *hoken* | 者 *sha* | 期間 *kikan* | を *wo* | 計算 *keisan* | する *suru* | 場合 *baai* | に *ni* | は *wa* | 、 | 月 *tsuki* | による *niyoru* | もの *mono* | と *to* | する *suru* | 。 |
| Original Tag | B-A | I-A | I-A | I-A | I-A | I-A | I-A | I-A | I-A | I-A | I-A | B-C | I-C | I-C | I-C | I-C | I-C |
| Bunsetsu | 1 | | | | | 2 | | 3 | | | | 4 | | 5 | | 6 | |
| New Tag | B-A | | | | | I-A | | I-A | | | | B-C | | I-C | | I-C | |

Figure 5: New setting for RRE task.

hinge-loss function and the margin of learning data.

1 Input: $S = (x_i; y_i), i = 1, 2, ..., n$ in which $x_i$ is the sentence and $y_i$ is a sequence of labels
2 Output: the model
3 Initialize: $w_1 = (0, 0, ..., 0)$
4 **for** $t$=*1, 2...* **do**
5 Receive an sentence $x_t$
6 Predict $y_t^* = \arg\max_{y \in Y}(w_t.\Phi(x_t, y_t))$
Suffer loss: $l_t = w_t.\Phi(x_t, y_t^*) - w_t.\Phi(x_t, y_t) + \sqrt{\rho(y_t, y_t^*)}$
7 Set:$\tau_t = \frac{l_t}{||\Phi(x_t, y_t^*) - \Phi(x_t, y_t)||^2}$
8 Update:
$w_{t+1} = w_t + \tau_t(\Phi(x_t, y_t) - \Phi(x_t, y_t^*))$
9 **end**

**Algorithm 1**: The Passive-Aggressive algorithm for RRE task.

Algorithm 1 shows briefly the Online Learning for sequence learning problem. The detail about this algorithm can be referred to the work of (Crammer et al, 2006). In Line 7, the argmax value is computed by using the Viterbi algorithm. Algorithm 1 is terminated after $T$ rounds [3]

#### 2.2.3 Feature Set

In the previous setting (Ngo et al., 10), we model the RRE task as a sequence labeling problem in which elements of sequences are words. Because a sentence may contain many words, the length of a sequence becomes large. Our idea is that, instead of considering words as elements, we consider each Bunsetsu as an element. This can be done because no Bunsetsu can belong to two different parts in this task. By doing this, we can reduce the length of sequences significantly. The process of obtaining a new setting from the previous one is illustrated in Figure 5.

In this example, the length of the sequence is reduced from 17 to 6. On average, in the Japanese National Pension Law corpus, the length of a sequence with the old setting (words) is **47.3**, while only **17.6** with the new setting (Bunsetsus).

We use features including head words, functional words, punctuations, and the co-occurrence of head words and functional words in a window size 1. A window size 1 in this model will cover three Bunsetsu. So, it is much longer than a window size 2 (which covers five words) in a model based on words. This is the reason why a window size 1 is sufficient in this model. The results show that modeling based on Bunsetsu, an important unit in Japanese sentences, is suitable for the RRE task.

There are two reasons that may explain why the Bunsetsu-based model is better than the word-based model. The first reason is that Bunsetsus are basic units in analyzing Japanese (in fact, dependency parsing of Japanese based on Bunsetsus, not words). Bunsetsus convey the meaning of a sentence better than words. In the Bunsetsu-based model, we only use head words and functional words to represent a Bunsetsu. Hence, the Bunsetsu-based model also take advantages of the model using head words and functional words. The second reason is that the Bunsetsu-based model reduces the length of sequences significantly compared with the word-based models. It helps the Bunsetsu-based model so much in the learning process. We choose the IOE strategy for our RRE task because this representation attained highest results on our development set (Ngo et al., 10).

## 3 A Simple Semi-Supervised Learning Method for RRE

This section describes a brief survey of semi-supervised learning, and presents a simple semi-supervised method for the RRE task using Brown word clusters (Brown et al., 92).

### 3.1 Brown Clustering

The Brown clustering algorithm is a word clustering algorithm based on the mutual information of bigrams (Brown et al., 92). The input to the algorithm is a set of words and a text corpus. In the initial step, each word belongs to its own individual clus-

[3] T is set to 10 in our experiments.

ter. The algorithm then gradually groups clusters to build a hierarchical clustering of words.

Figure 6 shows an example of Brown word-cluster hierarchy in a binary tree style. In this tree, each leaf node corresponds to a word, which is uniquely identified by the path from the root node to it. This path can be represented by a bit string, as shown in Figure 6. From the root node, we add bit 0 to the left branch and bit 1 to the right branch. A word-cluster hierarchy is reduced to depth $n$ if all words with the same $n$-bit prefix are grouped in one cluster. For example, if the word-cluster hierarchy in Figure 6 is reduced to depth 2, we will obtain a new hierarchy in Figure 7.

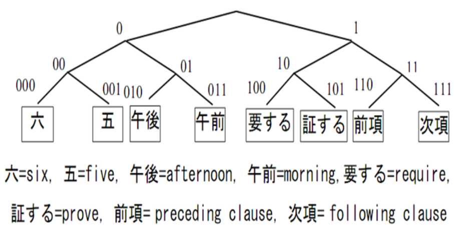


Figure 6: An example of Brown word-cluster hierarchy.

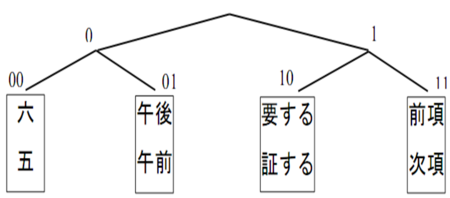


Figure 7: A Brown word-cluster hierarchy after reduction to depth 2.

Features extracted at $n$-bit depth are binary strings with length $n$. By reducing the word-cluster tree to different values of depth $n$, we can group words at various levels, from coarse clusters (small value of $n$) to fine clusters (large value of $n$).

### 3.2 RRE with Extra Word Features

The main idea of our semi-supervised learning method is to use *unsupervised* word representations as extra word features of a *supervised* model. We use Brown word clusters as the word representation method. In this framework, *unlabeled data* are used to produce word clusters. From these word clusters, we extract extra word features, and add these features to a *supervised* model (*labeled data* are used to train this model). Figure 8 shows our semi-supervised learning framework. This framework consists of two phase: *unsupervised* phase with the Brown clustering algorithm, and *supervised* phase with CRFs.

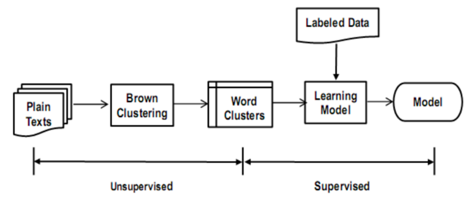


Figure 8: Semi-supervised learning framework.

To produce word representations, we first collected plain text from the address http://www.japaneselawtranslation.go.jp[4]. Our plain text corpus includes more than 13 thousand sentences about Japanese laws. After word segmenting (using Cabocha tool [5]), we conducted the Brown clustering algorithm to cluster words. In our work, we used the implementation of Percy Liang (Liang 05), and the number of clusters was set to 200. Experimental results showed that CRFs-LBFG obtained highest result in both supervised and semi-supervised experiments. In order to consider the impact of learning size to RRE tasks, we do an experiment with various size of training data.

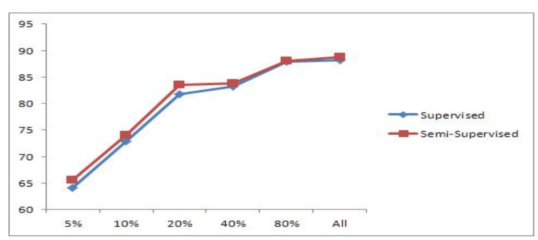


Figure 9: Comparison between the supervised method and the semi-supervised method with respect to the training sizes and F-measure scores.

The result in Fig 9 clearly showed that semi-supervised models are useful for RRE tasks with respect to various size of training data.

[4]This website provides many Japanese law articles in both Japanese and English.

[5]http://chasen.org/ taku/software/cabocha/

## 4 Experimental Results

We test our system on the corpus of Japanese National Pension Law, using F-measure for evaluation.

### 4.1 Corpus and Evaluation Method

This sub-section presents our corpus for the RRE task and evaluation method.The Japanese National Pension Law corpus includes 764 annotated Japanese law sentences[6]. Some statistics on this corpus are shown in Table 1. We have some remarks to make here. First, about 98.5% of sentences belong to the implication type, and only 1.5% of sentences belong to the equivalence type. Second, about 83.5% of topic parts are $T_2$, 15.2% of topic parts are $T_3$, and only 1.3% of topic parts are $T_1$. Finally, four main types of parts, $C$, $A$, $T_2$, and $T_3$ make up more than 98.3% of all types.

### 4.2 Evaluation Method

We divided the corpus into 10 sets and performed 10-fold cross-validation tests. The results were evaluated using precision, recall, and $F_{\beta=1}$ scores as follows:

$$precision = \frac{\#\text{correct parts}}{\#\text{predicted parts}}, recall = \frac{\#\text{correct parts}}{\#\text{actual parts}} \tag{3}$$

$$F_{\beta=1} = \frac{2 * precision * recall}{precision + recall} \tag{4}$$

A logical part is recognized correctly if and only if it has correct start word, correct end word, and correct part category (kind of logical part).

### 4.3 Experimental Results

Table 1 shows the comparison of three discriminative learning methods for RRE tasks. Three sequence learning methods include: CRFs using the LBFGS method, CRFs with SGD, and Online Learning. Experiment results show that the CRFs-LBFGS is the best in comparison with others. However, the computational times for training is slower than either SGD or Online Learning. The SGD is faster than CRF-LBFS approximately 6 times.

[6]The corpus consists of only the first sentence of each article.

Note that we used CRF++ [7] for Conditional Random Fields using LBFGS, and for Stochastic Gradient Descent (SGD) we used SGD1.3 which is developed by Leon Bottou [8].

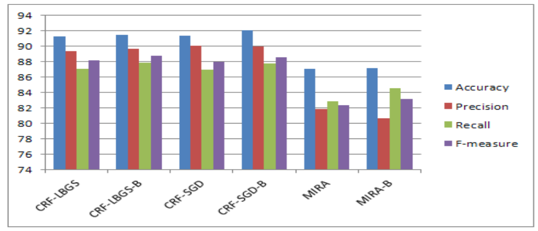


Figure 10: Comparison between the supervised method and the semi-supervised method.

For the RRE task, we extracted features at 4-bit depth and 6-bit depth. We integrated these features into three sequence learning models: CRFs-LBFG, CRF-SGD, and online learning (MIRA). The experimental results of the semi-supervised method with extra word features are shown in Figure 10. In three models, the semi-supervised method outperforms the supervised method. For CRF-LBFG model, the $F_{\beta=1}$ score was 88.80%, compared with 88.18% for the supervised method. The CRF-sgd model got 88.58% in the $F_{\beta=1}$ score, compared with 88.03% for the supervised method. MIRA method got 82.3% and 83.21% for supervised and semi-supervised models. In conclusion, word cluster models significantly improve the performance of sequence learning models for RRE tasks. We believe that word cluster models are also suitable for other sequence learning models.

## 5 Conclusions

In this paper, we report an investigation of developing a RRE task using discriminative learning models and semi-supervised models. Experimental results using 10 Folds cross-validation test have showed that the discriminative models are well suitable for RRE task. Conditional random fields show a better performance in comparison with other methods. In addition, word cluster models are suitable for improving the performance of sequence learning models for RRE tasks.

[7]http://crfpp.sourceforge.net/

[8]http://leon.bottou.org/projects/sgd

Table 1: Statistics on the Japanese National Pension Law corpus.

| **Sentence Type** | **Number** | **Part Type** | **Number** |
|---|---|---|---|
| Equivalence | 11 | $EL$ | 11 |
| | | $ER$ | 11 |
| Implication | 753 | **C** | **745** |
| | | **A** | **429** |
| | | $T_1$ | 9 |
| | | $\mathbf{T}_2$ | **562** |
| | | $\mathbf{T}_3$ | **102** |

Table 2: Experimental results with sequence learning models for RRE task.

| Methods | Accuracy | Precision | Recall | F-measure |
|---|---|---|---|---|
| CRF-LBFG | 91.27 | 89.328% | 87.039% | 88.158 |
| CRF-LBFG-B | 91.45 | 89.708% | 87.866% | 88.807 |
| CRF-SGD | 91.39 | 90.046% | 86.953% | 88.023 |
| CRF-SGD-B | 92.041 | 90.011% | 87.787% | 88.584 |
| MIRA | 87.081 | 81.881% | 82.909% | 82.339 |
| MIRA-B | 87.139 | 80.679% | 84.59% | 83.213 |

There are still room for improving the performance of RRE tasks. For example, more attention on features selection is necessary. We would like to solve this in future work.

## Acknowledgments

The work on this paper was partly supported by the grants for Grants-in-Aid for Young Scientific Research 22700139.

## References

K. Crammer et al. 2006. Online Passive-Aggressive Algorithm. Journal of Machine Learning Research, 2006

P.F. Brown, P.V. deSouza, R.L. Mercer, V.J.D. Pietra, J.C. Lai. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, Volume 18, Issue 4, pp.467-479, 1992.

X. Carreras, L. Màrquez, J. Castro. Filtering-Ranking Perceptron Learning for Partial Parsing. *Machine Learning*, Volume 60, pp.41-71, 2005.

M. Collins. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of EMNLP*, pp.1-8, 2002.

T. Katayama. Legal engineering - an engineering approach to laws in e-society age. In *Proceedings of the 1st International Workshop on JURISIN*, 2007.

T. Kudo, K. Yamamoto, Y. Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of EMNLP*, pp.230-237, 2004.

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

P. Liang. Semi-Supervised Learning for Natural Language. *Master's thesis, Massachusetts Institute of Technology*, 2005.

M. Murata, K. Uchimoto, Q. Ma, H. Isahara. Bunsetsu identification using category-exclusive rules. In *Proceedings of COLING*, pp.565-571, 2000.

Ngo Xuan Bach, Nguyen Le Minh, Akira Shimazu. Recognition of Requisite Part and Effectuation Part in Law Sentences. In Proceedings of (ICCPOL), pp. 29-34, San Francisco California USA, July 2010

M. Nakamura, S. Nobuoka, A. Shimazu. Towards Translation of Legal Sentences into Logical Forms. In *Proceedings of the 1st International Workshop on JURISIN*, 2007.

E.T.K. Sang, S. Buchholz. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL*, pp.127-132, 2000.

K. Tanaka, I. Kawazoe, H. Narita. Standard structure of legal provisions - for the legal knowledge processing by natural language. In *IPSJ Research Report on Natural Language Processing*, pp.79-86, 1993.

V.N. Vapnik. Statistical learning theory. New York: Wiley, pp.339-371, 1998.