Artif Intell Law
https://doi.org/10.1007/s10506-018-9225-1

# Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts

**Truong-Son Nguyen[1,2] · Le-Minh Nguyen[1] · Satoshi Tojo[1] · Ken Satoh[3] · Akira Shimazu[1]**



**Abstract** This paper proposes several recurrent neural network-based models for recognizing requisite and effectuation (RE) parts in Legal Texts. Firstly, we propose a modification of BiLSTM-CRF model that allows the use of external features to improve the performance of deep learning models in case large annotated corpora are not available. However, this model can only recognize RE parts which are not overlapped. Secondly, we propose two approaches for recognizing overlapping RE parts including the cascading approach which uses the sequence of BiLSTM-CRF models and the unified model approach with the multilayer BiLSTM-CRF model and the multilayer BiLSTM-MLP-CRF model. Experimental results on two Japan law RRE datasets demonstrated advantages of our proposed models. For the Japanese National Pension Law dataset, our approaches obtained an $F_1$ score of 93.27% and exhibited a significant improvement compared to previous approaches. For the Japan Civil Code RRE dataset which is written in English, our approaches produced an $F_1$ score of 78.24% in recognizing RE parts that exhibited a significant

✉ Le-Minh Nguyen
nguyenml@jaist.ac.jp

Truong-Son Nguyen
nguyen.son@jaist.ac.jp

Satoshi Tojo
tojo@jaist.ac.jp

Ken Satoh
ksatoh@nii.ac.jp

Akira Shimazu
shimazu@jaist.ac.jp

[1] Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan

[2] University of Science, VNU-HCMC, Ho Chi Minh city, Vietnam

[3] National Institute of Informatics, Tokyo 100-0003, Japan

Published online: 24 March 2018

improvement over strong baselines. In addition, using external features and in-domain pre-trained word embeddings also improved the performance of RRE systems.

**Keywords** Deep learning · Recurrent neural networks · Long short-term memory · Conditional random fields · Legal text analysis · Recognizing requisite and effectuation parts · Sequence labeling

## 1 Introduction

Analyzing legal texts is one of the essential tasks to understand the meaning of legal documents because it enables us to build information systems in the legal domain that assists people to exploit the information in legal documents effectively or check the contradiction and conflict in legal texts.

Unlike documents such as online news or users' comments in social networks, legal texts have special characteristics. Legal sentences are long, complicated and usually represented in specific structures. In almost all cases, a legal sentence can be separated into two main parts: a requisite part and an effectuation part. Each is composed of smaller logical parts such as antecedent, consequent, and topic parts (Nakamura et al. 2007; Tanaka et al. 1993). Depending on the granularity levels of the annotation scheme, an overlap between requisite and effectuation parts in law sentences might exist. The structure of law sentences is described in detail in Sect. 2.

Recognizing requisite and effectuation parts in legal texts, called the RRE task, can be modeled as a sequence labeling problem which can be solved by utilizing various kinds of models invented for this task. One such model is Conditional Random Fields (CRFs) employed by Ngo et al. (2010) to recognize RE parts in Japanese National Pension Law documents. The authors utilized a number of linguistic features such as headwords, function words, punctuations and Part-of-Speech features. The authors also applied a re-ranking model which used a linear score function to re-rank $k$-best outputs from CRFs. Later, Nguyen et al. (2011) improved the results using the Brown algorithm, an unsupervised learning model, to extract word cluster features on a large dataset. These features were then used to train models using supervised learning models including CRFs and the Margin-infused relaxed algorithm. However, these approaches only focused on recognizing non-overlapping RE parts. Consequently, if RE parts overlap, there is not a unified model that can recognize them.

Our work is motivated by the development in recent years of deep learning models. Many powerful deep learning models have been invented for solving a variety of Natural Language Processing (NLP) tasks such as machine translation, question answering, textual entailment and text categorization. In the sequence labeling task, a kind of text categorization, deep learning models show extremely performance on many tasks such as Part-of-Speech tagging (Wang et al. 2015b), Named Entity Recognition, chunking (Lample et al. 2016; Huang et al. 2015; Chiu

and Nichols 2015), semantic role labeling (Zhou and Xu 2015). The advantage of deep learning models is that we do not have to design feature sets because they contain different hidden layers which learn implicit features automatically and efficiently when the training corpus is large enough. However, in small datasets, feature sets can provide many benefits that improve the performance of deep learning models because they can provide new knowledge such as syntactic or semantic information. In addition, the design of deep learning models is very flexible in the sense that the same kind of a deep learning model can be adapted to different tasks. For example, a recurrent neural network can be used for different tasks such as image captioning, machine translation, sentiment analysis, and sequence labeling (Karpathy 2015).

This paper proposes several approaches that utilize deep learning models to recognize RE parts in legal documents. Firstly, we propose a modification of BiLSTM-CRF that allows the integration of external features to recognize non-overlapping RE parts more efficiently. Secondly, we propose two approaches including the cascading approach and the unified model approach for recognizing overlapping RE parts by modeling the task of RRE as a multilayer sequence labeling task. In the cascading approach, we recognize labels in all layers ($n$ layers) using a sequence of $n$ separate BiLSTM-CRF models in which each model is responsible for recognizing labels at each layer and these labels are then used as features for predicting labels at higher layers. This approach is inconvenient in training and predicting because we have to train many single models. Therefore, in the unified model approach, we propose two multilayer models, called the multilayer BiLSTM-CRF and the multilayer BiLSTM-MLP-CRF, which can recognize labels of all layers at the same time.

Experimental results on two Japanese RRE datasets showed that our model outperforms other approaches. On the Japanese National Pension Law RRE dataset, our models produced 93.27% in F1 score that exhibited a significant improvement compared to previous works. In the Japanese Civil Code RRE dataset, our proposed models outperform Conditional Random Fields on the same feature sets. The best model produced an F1 score of 78.24%. In two multilayer models, the multilayer BiLSTM-MLP-CRF is an improvement of the multilayer BiLSTM-CRF because it eliminates redundant components. Consequently, the training, testing time and the size reduce significantly but the performance is still competitive.

The remainder of the paper is organized as follows. First, Sect. 2 presents the typical structures of law sentences and the RRE task. Section 3 presents the background of several recurrent neural network-based models that are designed for sequence labeling tasks. Section 4 describes our proposed models for recognizing non-overlapping and overlapping requisite and effectuation parts. Section 5 describes our experiments including datasets, experimental settings, results and some discussions. Finally, our conclusions and future work are described in Sects. 6 and 7.

## 2 The RRE task

### 2.1 Structure of legal sentences

One of the most important characteristics of legal texts is that their sentences are presented in specific structures. In most cases, a legal sentence can roughly be divided into two parts: a requisite part and an effectuation part (Nakamura et al. 2007; Tanaka et al. 1993). These two parts are used to create legal structures of law provisions in legal articles and these structures are usually presented in the form below:

$$\text{requisitepart} \Rightarrow \text{effectuationpart}.$$

In more detail, requisite and effectuation parts are constructed from one or more logical parts such as antecedence parts, consequence parts, and topic parts. A logical part is a clause or phrase in law sentences at the lower level that contains a list of consecutive words. Each logical part carries a specific meaning of legal texts according to its type. A consequent part describes a law provision, an antecedent part describes cases or the context in which the law provision can be applied, and a topic part describes subjects related to the law provision (Ngo et al. 2013).

Four typical relationships between logical structures and logical parts are illustrated in Fig. 1 (Ngo et al. 2010). In the simple case (case 0), the requisite part only consists of one antecedent part (A) and the effectuation part only consists of one consequent part (C). In other cases, requisite parts and effectuation parts can

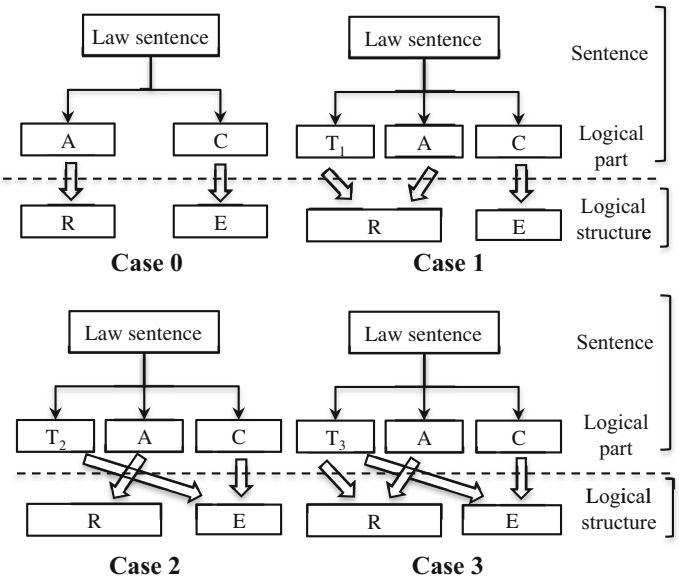


**Fig. 1** Four cases of the logical structure of a law sentence. *A* represents a antecedent part, *C* represents a consequent part, $T_i$ represents a topic part (Ngo et al. 2010)

**Table 1** Examples of overlapping and non-overlapping between requisite and effectuation parts in JCC-RRE dataset

| # | Original sentence | Requisite and effectuation parts |
|---|---|---|
| 1 | A child affiliated by his/her parents while they are married shall acquire the status of a child in wedlock from the time of that affiliation . | **R**: **A child** affiliated by his/her parents while they are married<br>**E**: **A child** shall acquire the status of a child in wedlock from the time of that affiliation *(overlapped part: A child - Case 3)* |
| 2 | A juristic act which is subject to a condition subsequent shall become ineffective upon fulfillment of the condition . | **R**: **A juristic act** which is subject to a condition subsequent<br>**E**: **A juristic act** shall become ineffective upon fulfillment of the condition. *(overlapped part: A juristic act - Case 3)* |
| 3 | If the party manifests an intention to extend the effect of fulfillment of the condition retroactively to any time prior to the time of the fulfillment , such intention shall prevail . | **R**: If the party manifests an intention to extend the effect of fulfillment of the condition retroactively to any time prior to the time of the fulfillment<br>**E**: such intention shall prevail *(non-overlapped - Case 1)* |
| 4 | If a person with limited capacity manipulates any fraudulent means to induce others to believe that he/she is a person with capacity , his/her act may not be rescinded . | **R**: If a person with limited capacity manipulates any fraudulent means to induce others to believe that he/she is a person with capacity<br>**E**: his/her act may not be rescinded *(non-overlapped - Case 0)* |

consist of two logical parts. In case 1, the requisite part consists of one antecedent part and one topic part (T1) that depends on the antecedent part. In case 2, the effectuation part consists of one consequent part and one topic part (T2) that depends on the consequent part. Case 3 shows the most complex form of a legal sentence in which the topic part depends both antecedent and consequent part, so this topic part (T3) will appear in both requisite and effectuation parts (Ngo et al. 2010). Tables 1 and 2 show some examples in our experimental datasets.

### *2.1.1 Non-overlapping and overlapping RRE datasets*

Because RE parts can be constructed from logical parts or from a list of individual words, we can create an RRE dataset in two following approaches. In the first approach, we annotate RE parts by annotating logical parts such as A, C, T1, T2, T3. The RRE task in these datasets is easier because there is non-overlapping between logical parts. The JPL-RRE dataset is annotated in this way, all RE parts in four structures are annotated by annotating logical parts (see some examples in Table 2). However, in the second approach, if RE parts are represented by a list of individual words, they might be overlapped. For example, in sentences 1 and 2 of Table 1, the requisite parts and the effectuation parts share some common words (*A child* or *A juristic act*). The appearance of overlapped parts causes some difficulties because most of the current machine learning approaches only consider non-overlapping RE parts. Our approaches focus on both of these two types of datasets by modeling the non-

**Table 2** Examples of non-overlapping between requisite and effectuation parts in JPL-RRE dataset. Tags A, C, Ti denote antecedence, consequence and topic parts. The dataset is in Japanese, but we include an English translation in each example

| # | Sentence annotated by logical parts | RE parts |
|---|---|---|
| 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><br><A> When a period of an insured is calculated, </A><br><C> it is based on a month. </C> | R: A<br>E: 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><br><T1>者については、</T1><br><C>前後の被保険者期間を合算する</C><br><T1> For the person </T1><br><A> who is qualified for the insured after s/he was disqualified, </A><br><C> the terms of the insured are added up together. </C> | R: T1 & A<br>E: C |
| Case 2 | <T2>年金給付は、</T2><A>その支給を停止すべき事由が生じたときは、</A><C>その事由が生じた日の属する月の翌月からその事由が消滅した日の属する月までの分の支給を停止する。</C><br><A> If grounds for suspending payment have arisen<A><br><T2>insurance benefits in pension form</T2><br><C>shall not be paid from the month following the month in which said grounds arose until the month in which the grounds cease to exist.</C> | R: A<br>E: T2 & C |

overlapping RRE task as single layer sequence labeling task and the overlapping RRE task as multilayer sequence labeling task. The details are presented in the next section.

## 2.2 RRE as single and multilayer sequence labeling tasks

The RRE task can be modeled as a sequence labeling task that recognize all logical parts in an input sentence by assigning tags into its words or phrases. Given an input sentence that contains a sequence of $l$ tokens (words or phrases), the RRE task recognizes RE parts by recognizing the tag of each token $s = \{w_1, w_2, \ldots, w_l\}$ using **IOB** notation.[1] In the **IOB** notation, tokens of a requisite or an effectuation part are annotated by I, B or O tags. The first token of a part is tagged by B−, remained tokens of this part are tagged by I− while tokens that do not belong any part are tagged by O−.

If RE parts do not overlap, we can organize them in one layer and treat them as a single layer sequence labeling task because each token will be assigned only one tag. However, if they overlap, we cannot consider the RRE task as a single layer sequence labeling task because each token may belong more than one part. In this case, RE parts are organized into different layers to avoid the overlapping and the RRE task is considered as a multilayer sequence labeling task. Table 3 shows several examples in non-overlapping and overlapping datasets. The details of deep learning models to recognize non-overlapping and overlapping RE parts are presented in Sect. 4.

[1] https://en.wikipedia.org/wiki/Inside_Outside_Beginning.

**Table 3** IOB notation in single and multiple layer RRE dataset

| Token | Layer 1 |
|---|---|
| 年金給付は、 | B-S2 |
| その | B-R |
| 支給を | I-R |
| 停止すべき | I-R |
| 事由が | I-R |
| 生じた | I-R |
| ときは、 | I-R |
| その | B-E |
| 事由が | I-E |
| 生じた | I-E |
| 日の | I-E |
| 属する | I-E |
| 月の | I-E |
| 翌月から | I-E |
| その | I-E |
| 事由が | I-E |
| 消滅した | I-E |
| 日の | I-E |
| 属する | I-E |
| 月までの | I-E |
| 分の | I-E |
| 支給を | I-E |
| 停止する。 | I-E |

(a) Non-overlapping REs in JPL-RRE data set

| Token | Layer 1 | Layer 2 |
|---|---|---|
| A | B-R | B-E |
| child | I-R | I-E |
| affiliated | I-R | O |
| by | I-R | O |
| his/her | I-R | O |
| parents | I-R | O |
| while | I-R | O |
| they | I-R | O |
| are | I-R | O |
| married | I-R | O |
| shall | O | I-E |
| acquire | O | I-E |
| the | O | I-E |
| status | O | I-E |
| of | O | I-E |
| a | O | I-E |
| child | O | I-E |
| in | O | I-E |
| wedlock | O | I-E |
| from | O | I-E |
| the | O | I-E |
| time | O | I-E |
| of | O | I-E |
| that | O | I-E |
| affiliation | O | I-E |
| . | O | - |

(b) Overlapping REs in JPC-RRE data set

In case (a), the dataset is annotated using the single layer approach because RE parts do not overlap. In case (b), the dataset is annotated using the multilayer approach because RE parts may overlap

## 3 Background: recurrent neural networks for sequence labeling tasks

### 3.1 Long short-term memory (LSTM)

Long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) is a variant of recurrent neural networks (RNNs), a kind of neural networks that can operate on sequential data, which is suitable for solving sequence labeling tasks.

Figure 2 shows the structure of an RNN (Elman 1990), which has an input layer $\mathbf{x}$, a hidden layer $\mathbf{h}$ and an output layer $\mathbf{y}$. In the sequence labeling task, $\mathbf{x} = (x_1, x_2, \ldots, x_l)$ represents input embedding vectors of a sequence of tokens and $\mathbf{y} = (y_1, y_2, \ldots, y_l)$ represents output tags where $l$ is the length of the input sentence. If a sequence is considered as a kind of time series data, each embedding vector $x_t \in \mathbb{R}^D$ represents features of the token at time $t$ ($\text{token}_\text{t}$). These could be one-hot-encoding vectors, dense vectors or sparse vectors. Firstly, each hidden state $\mathbf{h}_t \in \mathbb{R}^H$, which represents contextual information which learned from $x_t$ and the previous context, is computed from previous hidden states and $x_t$ (Eq. 1). Each $\mathbf{v}_t \in \mathbb{R}^T$, which represents the probability distribution over tags of $\text{token}_\text{t}$, will then be computed from $\mathbf{h}_t$ using the *softmax* activation function (Eq. 2). Finally, the

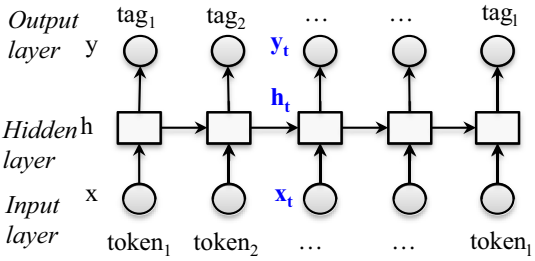


**Fig. 2** Recurrent neural networks

output tag $y_t \in [1, T]$ is obtained using *argmax* (Eq. 3). The values of the hidden and output layers of an RNN are computed as follows:

$$\mathbf{h}_t = f(\mathbf{U}x_t + \mathbf{W}\mathbf{h}_{t-1}) \tag{1}$$

$$\mathbf{v}_t = g(\mathbf{V}\mathbf{h}_t) \tag{2}$$

$$y_t = \arg\max_{i \in [1,T]} \mathbf{v}_{t_i} \tag{3}$$

where $D$, $H$ is the size of the input and hidden layers, $T$ is the number of tags in the tag set and the size of the output layer, $\mathbf{U}_{H\times D}$, $\mathbf{W}_{H\times H}$, and $\mathbf{V}_{T\times H}$ are the connection weights to be computed in training time, $f(z)$ and $g(z)$ are *sigmoid* and *softmax* activation functions as follow:

$$\begin{aligned} f(z) &= \frac{1}{1+e^{-z}} \\ g(z_m) &= \frac{e^{z_m}}{\sum_k e^{z_k}} \end{aligned} \tag{4}$$

RNNs, in theory, can capture the long range of dependencies, but they fail in practice due to the gradient vanishing / exploding problem (Bengio et al. 1994), this is one big limitation of RNNs. LSTMs solve this limitation by incorporating a memory cell that is able to capture long-range dependencies. They incorporate several gates that control the proportion of the input to the memory cell, and the proportion of the previous state to forget (Hochreiter and Schmidhuber 1997). The memory cell and gates can be implemented in different ways which are described in detail in Greff et al. (2017). Below is an implementation in Greff et al. (2017) which is used in Lample et al. (2016) and our research:

$$\begin{aligned} \mathbf{i}_t &= \sigma\left(\mathbf{W}_{xi}x_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i\right) \\ \mathbf{f}_t &= 1 - \mathbf{i}_t \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh\left(\mathbf{W}_{xc}x_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c\right) \\ \mathbf{o}_t &= \sigma\left(\mathbf{W}_{xo}x_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o\right) \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right) \end{aligned}$$

where $\sigma$ is the logistic sigmoid function, and $\odot$ is the element-wise product. **i**, **f**, **o** and **c** are the input gate, forget gate, output gate and memory cell vectors. $\mathbf{W_{ij}}$ matrices are connection weights that will be updated to minimize the loss function in training time. Below is the cross-entropy loss function that measures the difference between the output tags of the model and the real tags.

$$loss = -\frac{1}{l}\sum_{t=1}^{l}\sum_{i=1}^{T} \text{ygold}_{t_i} \log(\mathbf{v}_{t_i}) \tag{5}$$

where $\text{ygold}_t \in \mathbb{N}^T$ is the one hot vector which represent the true tag of $\text{token}_t$.

### 3.2 Bidirectional Long short-term memory (BiLSTM)

The LSTM mentioned in the previous section is also called *the forward LSTM* because it predicts the label of the current time based on the previous information. For example, in sequence labeling tasks of NLP, a forward LSTM will predict the label of a token based on the knowledge learned from previous tokens. However, the relationship between words in a sentence is bidirectional, so the label of a current token may be affected by tokens from both sides of this token. Therefore, the combination of a forward and backward LSTM, called BiLSTM (Graves et al. 2013), enables the model to learn both past and future information to predict the label at the current time. Figure 3 shows the architecture of a BiLSTM in which the hidden state $\mathbf{h}_t$ represented for knowledge learned from the $\text{token}_t$ and its context is the concatenation of forward and backward hidden states.

### 3.3 BiLSTM-CRF

BiLSTM-CRF is a combination of BiLSTM and Conditional Random Fields, a strong algorithm for sequence labeling tasks. CRFs will take a sequence of tokens and produce a sequence of tags that maximizes the log conditional probability of the output tag sequence given an input. CRFs are more effective than Hidden Markov models and Maximum Entropy (Lafferty et al. 2001) and show their effectiveness in a variety of sequence labeling tasks such as named entity recognition in biomedical texts (Settles 2004), morphological analysis (Kudo et al. 2004), shallow parsing (Sha and Pereira 2003), etc.

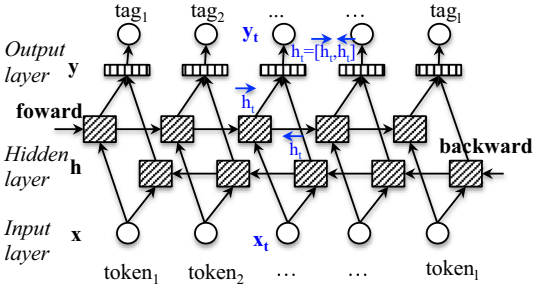


**Fig. 3** Bidirectional Long short-term memory model

In an LSTM or a BiLSTM model, the tagging decision of a token at the output layer is performed independently using the *softmax* activation function based on the hidden state of that token. This means that the final tagging decision of a token is local because it does not depend on the tagging decision of others. Therefore, adding a CRF layer into an LSTM or a BiLSTM will make the tagging decision global. In other words, the model can learn to find best tag sequence in all possible output tag sequences. This model is described in detail in Huang et al. (2015), Lample et al. (2016), Wang et al. (2015a) and Wang et al. (2015b).

Assume that $P$ is the matrix of scores output by the bidirectional LSTM components. P is of size $l \times k$, where k is the number of distinct tags, and $P_{ij}$ corresponds to the score of the $j$ th tag of the $i$ th word in a sentence. For a sequence of predictions $\mathbf{y} = (y_1, y_2, \ldots, y_l)$, its score is defined by:

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{l} A_{y_i, y_{i+1}} + \sum_{i=1}^{l} P_{i, y_i} \tag{6}$$

where $A$ is a matrix of transition scores such that $A_{ij}$ represents the score of a transition from the tag $i$ to tag $j$. Because tags $y_0$ and $y_{l+1}$ indicate the start and the end a sentence, $A$ is therefore a square matrix of size $k + 2$. A probability for the sequence $\mathbf{y}$ over all possible tag sequences will then be calculated using a *softmax*:

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X}, \mathbf{y})}}{\sum_{\tilde{y} \in \mathbf{Y_X}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})}} \tag{7}$$

During training, BiLSTM-CRF will maximize the log-probability (or minimize the negative of the log-probability) of the correct tag sequence:

$$\begin{aligned} \log(p(\mathbf{y}|\mathbf{X})) &= s(\mathbf{X}, \mathbf{y}) - \log\left(\sum_{\tilde{y} \in \mathbf{Y_X}} e^{s(\mathbf{X}, \tilde{y})}\right) \\ &= s(\mathbf{X}, \mathbf{y}) - \underset{\tilde{y} \in \mathbf{Y_X}}{\text{logadd}}(s(\mathbf{X}, \tilde{\mathbf{y}})) \end{aligned} \tag{8}$$

where $\mathbf{Y_X}$ represents all possible tag sequences for a sentence $\mathbf{X}$. While decoding, the output tag sequence is the one that has the maximum score in all possible tag sequences given by:

$$\mathbf{y}^* = \underset{\tilde{y} \in \mathbf{Y_X}}{\arg\max} s(\mathbf{X}, \tilde{\mathbf{y}}) \tag{9}$$

where Eqs. 8 and 9 can be calculated using a dynamic programming algorithm (e.g Viterbi).

*Training neural networks* A neural network can be trained using the back-propagation algorithm (Boden 2001). Firstly, parameters/weights of the neural network are initialized randomly. They will then be updated through time to optimize the objective function (the cross-entropy loss or the log-probability ) using popular methods such as Stochastic Gradient Descent (Bottou 2010). In addition, the dropout technique (Srivastava et al. 2014) may be applied to avoid the over-

fitting and a validation set may be used to choose the optimum parameters or to decide when the training process stops.

*Initializing of word embedding vectors* Using pre-trained word embedding vectors is a way to improve the performance of the system. The embedding vector of each word is obtained from a lookup-table which can be initialized randomly or from pre-trained embedding sources. Word embeddings in a pre-trained source can be learned using different models such as word2vec (Mikolov et al. 2013; Ling et al. 2015), GloVe (Pennington et al. 2014) or fastText (Bojanowski et al. 2017). These embedding vectors then can be continually optimized in the training phase as other parameters.

*The effectiveness of BiLSTMCRF* BiLSTM-CRF has shown great success in Named Entity Recognition task without using any engineering features (Lample et al. 2016). However, when this model was applied to other NER datasets (e.g. Vietnamese NER dataset), this model could not outperform other conventional classifiers such as Conditional Random Fields with a set of engineering features (Nguyen et al. 2016b). We consider that because the size of the datasets is small, the network cannot obtain enough information to train a good model. In addition, in the multilayer tagging task, recognizing labels of a higher layer is affected by the recognized labels of lower layers, so the model should utilized labels of previous layers as the input features. However, the design of BiLSTM-CRF in Lample et al. (2016) cannot recognize labels in multilayer datasets.

In the legal domain, BiLSTM-CRF were also employed for analyzing legal texts in Vietnamese legal documents. Nguyen et al. (2016a) employed the BiLSTM-CRF to recognize RE parts in Vietnamese legal documents. The method exhibited a little improvement compared to CRFs (Nguyen et al. 2015). However, the approach did not use any features except the headwords of input sentences.

Due to above limitations, the next section will present four our proposed models which are based on BiLSTM-CRF to deal with the task RRE in legal texts. First, we proposed the single BiLSTM-CRF with features to recognize non-overlapping RE parts. Second, we proposed three models to recognize overlapping RE parts including the sequence of BiLSTM-CRF, the multilayer BiLSTM-CRF and the multilayer BiLSTM-MLP-CRF model to recognize overlapping RE parts.

## 4 Models

### 4.1 The single BiLSTM-CRF with features to recognize non-overlapping RE parts

Figure 4 shows the architecture of BiLSTM-CRF with features. The input of BiLSTM-CRF model is a sequence of vectors. Each vector represents a word in the input sentence. In the original model, these vectors are word embedding vectors representing only headwords. However, assume that each word/token in the input sentence has several features such as Part-of-speech, Chunk beside the headword.

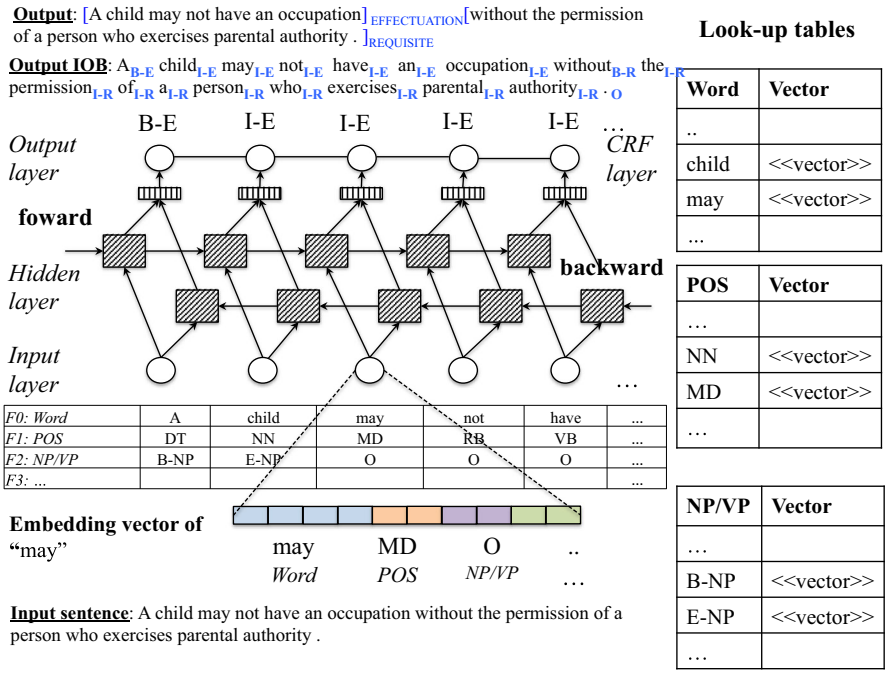


**Fig. 4** A Single BI-LSTM-CRF with features to recognize non-overlapping RE parts

The proposed model is achieved by adding some embedding layers that map those features into vector representations using look-up tables. The final vector representation of each word is obtained by concatenating the word embedding vector and all embedding vectors represent its features. For example, the final vector representation of the word "*may*" (Fig. 4) is the concatenation of the embedding vector of "*may*", and embedding vectors that represent its POS and chunk feature. While the look-up table of words can be initialized randomly or from a pre-trained source, look-up tables of features are initialized randomly.

The BiLSTM component then is used to encode the input sequence into hidden states where each of them represents knowledge that learns from each input word and its context. The hidden state vectors of the forward and backward LSTM represented for each input word then are concatenated into a single vector. This vector is then used to compute the tag score vector of the input word using another fully connected layer. If the CRF layer is used, these score vectors will then be used to find the best output tag sequence using the Viterbi decoding algorithm and a transition matrix learned from training process. Otherwise, the output tag of a token is obtained independently using the *argmax* function from a *softmax* of its tag score vector. Finally, requisite and effectuation parts will be constructed from the sequence of IOB tags (Fig. 4).

If the CRF layer is used, we use the negative of log-probability (Eq. 8) as the loss function. Otherwise, the cross-entropy loss (Eq. 5) is used to compute the loss of the

model during the training process. These objective functions are also used in Lample et al. (2016).

Algorithm 1 explains training and prediction procedure of the proposed model. The training procedure of the single BiLSTM-CRF with features, which is described in lines 1–21, includes several important steps such as feature extraction (line 4), forward the input through the network and update parameters (lines 12–13), evaluate and save the model if it improves the result on the validation set (lines 15–19). The parameter **externalFeatures** enables the use of this model for the cascading approach which is presented in 4.2. The prediction phase (lines 22–29) includes some important steps such as load the saved model (line 23), extract features of the input sentence (lines 24–25), create the input and predict the tag sequences of the input sentence (lines 26–28).

**Algorithm 1** Training and prediction procedure for a single BI-LSTM-CRF with features model

```
1: procedure TRAINSINGLE(Corpus, featureTypes, externalFeatures=None)
2:     inputs ← ∅
3:     for s ∈ Corpus do
4:         f ← extractFeature(s, featureTypes) ∪ externalFeatures
5:         inputs ← inputs ∪ (s, f)
6:     end for
7:     trainset, valSet ← divide(inputs)
8:     BiLSTMCrf ← createBiLSTMSCrf()
9:     performance ← 0
10:    for i ∈ 1..nEpoch do
11:        for input ∈ trainSet do
12:            BiLSTMCrf.forward(input)
13:            BiLSTMCrf.updateWeights()        ▷ Using back-propagation method with
    Stochastic Gradient Descent
14:        end for
15:        performance = BiLSTMCrf.evaluate()
16:        if performance > bestPerformance then
17:            BiLSTMCrf.saveModel()        ▷ Evaluate the model on the validation set, then
    save the model if it produces the better results on the validation set.
18:            bestPerformance ← performance
19:        end if
20:    end for
21: end procedure
22: procedure PREDICTSINGLE(s, model)
23:    BiLSTMCrf ← loadBiLSTMSCrf(model)
24:    featureTypes ← BiLSTMCrf.featureTypes
25:    f ← extractFeature(s, featureTypes, None)
26:    input = (s, f)
27:    tagSenquences ← BiLSTMCrf.predict(input)
28:    return tagSenquences
29: end procedure
```

## 4.2 The cascading approach to recognize overlapping RE parts

Recognizing overlapping RE parts can be viewed as a multilayer sequence labeling task mentioned in Sect. 2. We can simply train many models in which each model can predict tags at a certain layer. In the RRE task, the tag of a token at a layer may depend on tags at previous layers of this token. For example, in the JCC-RRE corpus, if the tag of a token in layer 1 is **B-E**, the tag of that token in layer 2 is

usually **B-R** (see the example in Table 3). Therefore, the model which predicts tags at a layer should use output tags of previous layers as features.

We propose a cascading approach that employs a sequence of BiLSTM-CRF models described in Sect. 4.1 to recognize RE parts in all layers. The training and prediction phases of the sequence of BiLSTM-CRF models is described in algorithm 2. In the training phase, we first determine $n$ as the number of layers in training corpus (line 2). The $i^{th}$ model in the sequence of $n$ BiLSTM-CRF models then is trained using word embeddings, features and tags of layer 1 to $i-1$ as external features (lines 4–8). In the prediction phase, to predict tags of layer $i$, we must predict tags of previous layers (1 to $i-1$) then use these tags for predicting tags of layer $i$ (lines 16–20). Finally, the output is the union of tags of all layers.

**Algorithm 2** Training and prediction of the multilayer tagging task using a sequence of BI-LSTM-CRF models

```
1: procedure TRAINSEQUENCE(Corpus, featureTypes)
2:     n ← number of layer in the training corpus
3:     for i ∈ 1..n do        ▷ Train a single BI-LSTM-CRF model m_i which is responsible to predict the tag at layer i^th
4:         if i = 1 then
5:             trainSingle(Corpus, featureTypes, None)                    ▷
6:         else
7:             tags ← tagsOfLayers(Corpus, [1, i − 1])
8:             trainSingle(Corpus, featureTypes, tags)        ▷ Using tags in layer 1-i − 1 as features to train the model i^th
9:         end if
10:    end for
11: end procedure
12: procedure PREDICTSEQUENCE(test, models)
13:    outputTagsOfAllLayers ← ∅
14:    n ← number of layer in the training corpus
15:    tagsOfPreviousLayers ← None
16:    for i ∈ 1..n do        ▷ Use model m_i and tags of layer i − 1 to predict tag at layer i
17:        tags ← predictSingle(test, models[i], tagsOfPreviousLayer)
18:        outputTagsOfAllLayers ← outputTagsOfAllLayers ∪ tags
19:        tagsOfPreviousLayers ← tagsOfPreviousLayers ∪ tags
20:    end for
21:    return outputTagsOfAllLayers
22: end procedure
```

### 4.3 Multi-BiLSTM-CRF to recognize overlapping RE parts

The use of separate models in the cascading approach to recognize overlapping RE parts is inconvenient for training and prediction because we must train separate models to recognize labels at different layers. For the prediction phase, we have to recognize labels of the lower layers then use these labels as features for predicting labels of higher layers. Therefore, we proposed a unified model that simplifies the training and prediction process because we train only one model to predict labels of all layers at the same time. The whole architecture of the model, called the multilayer BiLSTM-CRF or Multi-BiLSTM-CRF, is illustrated in Fig. 5.

This model is constructed from many BiLSTM-CRF components where each of them is responsible to predict labels of each layer. The input of a component at a

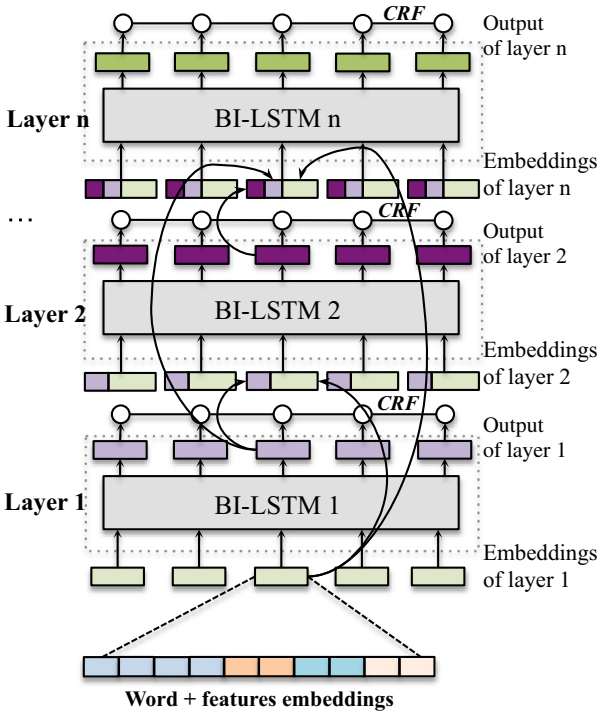


**Fig. 5** The multilayer BiLSTM-CRF model to recognize overlapping RE. Each BiLSTM-CRF component will predict tags at a layer and compute the tag score vectors for higher components

certain layer is a sequence of vectors in which each vector is the concatenation of word embedding, feature embeddings and tag score vectors of previous layers. The sequence of vectors then is used to compute the tag score vectors to predict tag at this layer and these vectors are used as features for higher layers.

$$loss = \sum_{i=1}^{nlayer} loss_i \tag{10}$$

The loss of Multi-BiLSTM-CRF models is computed from the loss of all its layers (Eq. 10). The loss of each layer is calculated in the same way as the loss of a BiLSTM-CRF model presented in Sect. 4.1. Multi-BiLSTM-CRF is also trained as a normal neural network which uses back-propagation and gradients to update network parameters that minimize the loss function.

### 4.4 Multi-BiLSTM-MLP-CRF to recognize overlapping RE parts

The advantage of Multi-BiLSTM-CRF mentioned in the previous section is that it possesses a convenient design that can simplify the training and prediction process.

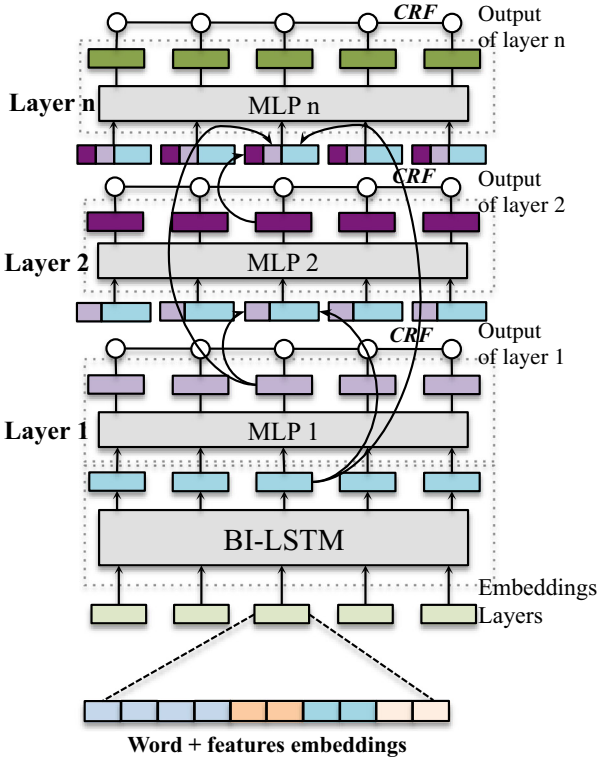


**Fig. 6** The multilayer BI-LSTM-MLP-CRF model to recognize overlapping RE. The BiLSTM component encodes the input sequence that produces hidden vectors for MLP components to predict the labels. Each MLP component of each layer will compute tag scores for higher layers and predict tags at that layer

Using this model, we can train only one model to predict labels at all layers. However, it also contains several limitations. Firstly, the number of parameters of the Multi-BiLSTM-CRF and all models in the sequence of BiLSTM-CRF (Sect. 4.2) are comparable. Consequently, the training time is not reduced significantly and the performance of these two models are similar. Secondly, in Multi-BiLSTM-CRF, the input sentence has been encoded many times in the same way by BiLSTM components. This may cause some ineffectivenesses in the training time and it contains redundant parameters. Due to those reasons, we propose an improvement of Multi-BiLSTM-CRF model, called Multi-BiLSTM-MLP-CRF, that eliminates redundant LSTM components thus it can reduce the training time and redundant parameters. The architecture of Multi-BiLSTM-MLP-CRF is illustrated in Fig. 6.

This model has only one BiLSTM component which is used to encode the input sentence into sequence of hidden states. These hidden states then will be used to predict output tags of $n$ layers using $n$ multilayer perceptron (MLP) components in

which each MLP is used to predict tags at each layer of the input sentence. The loss function of the model is also computed from the loss of all layers and it also trained using the back-propagation and gradients to update the parameters.

All proposed models are implemented using Python language and Theano library. Source codes of these models are available on Github.[2]

## 5 Experiments

### 5.1 Datasets and feature extraction

*The Japanese National Pension Law RRE dataset* (*JPL-RRE*) This dataset is in Japanese which is obtained from Ngo et al. (2010) and Nguyen et al. (2011). All sentences in JPL-RRE had been segmented into Bunsetsus chunks using the CaboCha tool (Taku Kudo 2002) and the tagging is at Bunsetsu level. The dataset had also been split into ten folds. In addition, some features such as headword, function words, punctuation marks, and word cluster features (Nguyen et al. 2011) had been extracted, so our study does not focus on the feature extraction step. In addition, this dataset is a non-overlapping dataset because it uses lower level parts (topic parts, antecedent, and consequent parts) to represent RE parts. Therefore, we can recognize RE parts in this dataset using a single BiLSTM-CRF (Sect. 4.1).

*Japanese Civil Code RRE dataset* (*JCC-RRE*) This dataset is the English translation version of the Japanese Civil Code which is annotated manually by three annotators supported by the CREST[3] project. This dataset contains three type of logical parts: requisite, effectuation parts and *Unless* parts. An *unless* part is a special part which describes an exception in law sentence. An *Unless part* usually begins with the word *"unless"* or *"provided, however"*. For example, the *unless* part of the below sentence is marked with **{...}**:

"[For acts where there is a conflict of interest between the assistant or his/her representative and a person under assistance]$_{\mathbf{R}}$, [the assistant shall apply to the family court for the appointment of a temporary assistant]$_{\mathbf{E}}$ ; **{**provided that [this shall not apply]$_{\mathbf{E}}$ [in the case where there is a supervisor of an assistant]$_{\mathbf{R}}$ **}**$_{\mathbf{U}}$".

Different from JPL-RRE dataset, RE parts in JCC-RRE may be overlapped. Therefore, RE parts in this dataset are organized in three different layers using the multilayer tagging approach. Examples in these two datasets are shown in Table 3. Sentences, features and RE parts in these two datasets are organized in the CoNLL format.

*Feature extraction for the JCC-RRE dataset* We use Stanford parser tools (Klein and Manning 2003) to parse all sentences in the corpus. We then extract a set of 5 syntactic features for the RRE task including POS tags, noun/verb phrases, relative clauses, clauses that begin with prepositions (e.g., "*if*", "*in cases*") and other subordinate clauses based on these syntactic parse trees. Values of these features are

[2] https://github.com/ntson2002/rre-tagging.

[3] https://www.jst.go.jp/kisoken/crest/en/.

**Table 4** An example of the feature exaction step in JCC-RRE dataset

| Features | If | an | heir | dies | without | having | made | acceptance | .. |
|---|---|---|---|---|---|---|---|---|---|
| POS | IN | DT | NN | VBZ | IN | VBG | VBN | NN | .. |
| Verb and noun phrase | – | B-NP | E-NP | – | – | – | – | B-NP | .. |
| Relative clause | – | – | – | – | – | – | – | – | .. |
| Clause begin with a preposition | B-If | I-If | I-If | I-If | I-If | I-If | I-If | I-If | .. |
| Subordinate clauses | – | – | – | – | – | – | – | – | .. |

We also use IOB notation to represent features. For example, we use B-NP, I-NP, E-NP to indicate the word is the begin, inside and the end of a noun phrase

**Table 5** The statistic of JPL-RRE and JCC-RRE datasets

| Type | Layer 1 | Layer 2 | Layer 3 | Description |
|---|---|---|---|---|
| Japanese Civil Code RRE | | | | |
| R | 2412 | 1 | 0 | Requisite part |
| E | 1410 | 676 | 0 | Effectuation part |
| U | 0 | 0 | 259 | Unless part |
| Japanese Pension Law RRE | | | | |
| E | 745 | – | – | Consequent parts in case 1, 2, 3 |
| R | 429 | – | – | Antecedent parts in case 1, 2, 3 |
| S1 | 9 | – | – | Topic part in case 1 |
| S2 | 562 | – | – | Topic part in case 2 |
| S3 | 102 | – | – | Topic part in case 3 |
| EL | 11 | – | – | Requisite part in case 0 |
| ER | 11 | – | – | Effectuation part in case 0 |

categorical values, and an example of these features is shown in Table 4. These features are expected to encourage the deep learning models to recognize the boundary of RE parts better. The statistics of these two datasets are shown in Table 5.

## 5.2 Evaluation methods

We use 10-fold cross validation with Precision, Recall, and F-measure ($F_1$) scores to evaluate our models. After training, the trained models are used to predict IOB labels of tokens of all sentences in test sets. These IOB labels are then used to construct RE parts. We employ the **conlleval** tool (Tjong Kim Sang and De Meulder 2003) to evaluate the performance of proposed models. This tool employs the strict matching method to evaluate the performance. That means an RE part is considered to be correct if and only if all its words are predicted correctly. *Precision*, *Recall* and $F_1$ scores are then calculated as follows:

$$precision = \frac{\#\text{correctparts}}{\#\text{predictedparts}}, \quad recall = \frac{\#\text{correctparts}}{\#\text{actualparts}} \tag{11}$$

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{12}$$

In addition, when we use the sequence of BiLSTM-CRF models to recognize labels in all layers of JCC-RRE dataset, recognizing labels at a layer is affected by recognizing labels previous layers. Therefore, these models can be evaluated using two following methods:

- *Single layer evaluation* In this method, the performance of each layer is conducted with the assumption that the labels at previous layers are totally correct.
- *End-to-end evaluation* The performance of each layer is conducted after using trained models to predict labels of previous layers. This evaluation method produces the overall performance of the system.

### 5.3 Experimental setting and design

We conducted experiments on both mentioned datasets. The experiments are designed to evaluate the performance of the models and find the best configurations. We also compared our models with several baselines. For the JPL-RRE dataset, we compare our models with the best result from experiments conducted by Ngo et al. (2010), Nguyen et al. (2011). Because the JCC-RRE dataset is new, we compared proposed models with CRF, a strong algorithm for sequence labeling tasks. We also examined different configurations to evaluate the effectiveness of, feature sets, pre-trained word embeddings, or different RNN-based models.

Tuning all hyper-parameters is time-consuming, we followed the recommendations from Lample et al. (2016) for choosing the value of hyper-parameters such as dropout rate = 0.5, word embedding size = 100, hidden size = 100. The embedding size for each feature is set to 10. Besides, we use the back-propagation method and the stochastic gradient descent algorithm to train our neural networks. For each experiment, each model is trained within 200 epochs and the parameters are saved when the model improves the performance of the validation set. The learning rate is set to 0.002 for all models when the CRF layer is used, otherwise learning rate is 0.01. In addition, we also use the **IOBES** tagging scheme instead of **IOB**. The **IOBES** tagging scheme is a variant of **IOB** tagging scheme in which the end token of a part is labeled by tag E- and RE parts which have only one token are labeled by tag S-.

*Pre-trained word embedding vectors for the legal domain* To train word embeddings for the legal domain, we crawl a collection of legal documents from the website of the Ministry of Justice, Japan.[4] Then, we use the word2vec model

[4] http://www.japaneselawtranslation.go.jp: This site contains the English translation of Japanese legal documents including Japanese Civil Code.

(Mikolov et al. 2013) to learn word embedding representations for RRE task. Currently, we only learn word embedding representations for words in the JCC-RRE corpus.

### 5.4 Results

*Results on the JPL-RRE dataset* Table 6 shows the performance of BiLSTM-CRF in the JPL-RRE datasets and four baselines (Nguyen et al. 2011; Ngo et al. 2010). Compared to the best baseline, our models exhibited significant improvements. Adding features into deep learning models also improved the performance of RRE systems. For example, the result increased by + 2.25% in $F_1$ score when headwords and function words were used. When punctuation features were used, the result increased by + 2.44% . Finally, when all feature sets were used, the model produced an $F_1$ score of 93.27% which increased by + 4.46% compared to the best baseline.

*Results on the JCC-RRE dataset* Table 7 shows the results of all models on the JCC-RRE dataset. The baseline (model 1) is the sequence of CRFs which is implemented using CRF++ (Kudo 2005) . Model 2 and 3 also apply the cascading approach with the use of the sequence of BiLSTM and BiLSTM-CRF models (Sect. 4.2). Last three models (4,5,6) are multilayer models which are described in Sects. 4.3 and 4.4. The number of fully connected layers in Multi-BiLSTM-MLP1-CRF and Multi-BiLSTM-MLP2-CRF are 1 and 2, respectively. A clear comparison is illustrated in Fig. 7. In addition, the detailed results of each label are presented in Table 8.

*Comparison with the baseline* With the same feature set, proposed models outperform the baseline significantly, except for the sequence of BiLSTM. For example, when only word features were used, the sequence of CRFs produced an $F_1$ of 70.8%, but all proposed models (3,4,5,6) without using pre-trained embeddings produce F1 scores from 73.95 to 75.31% that improves from 3 to 4%. If pre-trained embeddings were used, the performance is better, proposed models improved the baseline from 6 to 7% in $F_1$ score. That trend is the same when word and syntactic features were used; the proposed models improved the baseline from 1 to 2 and 3.5 to 4.5% in $F_1$ score depend on whether or not pre-trained embeddings were used.

*The effectiveness of a CRF layer in neural network models for RRE task* In two kinds of LSTM-based models, the use of the CRF layer improves the performance significantly. For example, in Table 7, the performance of the sequence of BiLSTM is less than BiLSTM-CRF from 13 to 14%. This result shows a strong relationship between tags of the output tag sequence. Therefore, adding a CRF layer, the model can find the best output tag sequence based on the transition between tags learned from the corpus.

*The effectiveness pre-trained embeddings and features* Using pre-trained embeddings always improved the performance of RRE systems, especially if the pre-trained embeddings are learned from the in-domain corpus. In all experiments, using

**Table 6** Experimental results on the Japanese National Pension Law RRE datasets with different feature sets

| Model + features | Precision (%) | Recall (%) | F1 (%) | | |
|---|---|---|---|---|---|
| CRF | | | | | |
| 1 | HW + FW + Punctuation | 88.09 | 86.30 | 87.19 | |
| 2 | BC (Bunsetsu) | 88.75 | 86.52 | 87.62 | |
| 3 | BC (Bunsetsu) + Reranking | 89.42 | 87.75 | 88.58 | |
| 4 | BC (Bunsetsu) + Brown cluster | 89.71 | 87.87 | 88.81 | Baseline |
| BI-LSTM-CRF | | | | | |
| 1 | HW + FW | 90.62 | 91.50 | 91.06 | + 2.25% |
| 2 | HW + FW + Punctuation | 91.05 | 91.45 | 91.25 | + 2.44% |
| 3 | HW + FW + Punctuation + Brown cluster | 92.77 | 93.77 | 93.27 | + 4.46% |

Results of CRF models (1–4) are from Ngo et al. (2010) and Nguyen et al. (2011)

**Table 7** Experimental results ($F_1$ score) on JCC-RRE dataset using end-to-end evaluation method

| Models | Pre. emb | Features | Layer 1 | Layer 2 | Layer 3 | All |
|---|---|---|---|---|---|---|
| (1) Sequence of CRFs (**baseline**) | | Word | 73.23 | 56.49 | 88.8 | 70.8 |
| | | Word+Syn. | 76.52 | 59.07 | 87.82 | 73.7 |
| (2) Sequence of BiLSTM | | Word | 61.35 | 53.60 | 85.33 | 60.96 |
| | | Word+Syn. | 62.02 | 54.72 | 80.60 | 61.49 |
| | x | Word | 64.17 | 56.56 | 82.71 | 63.58 |
| | x | Word+Syn. | 64.57 | 57.41 | 83.71 | 64.06 |
| (3) Sequence of BiLSTM-CRF | | Word | 76.37 | 61.41 | 88.20 | 73.95 |
| | | Word+Syn. | 77.22 | 61.02 | 90.27 | 74.65 |
| | x | Word | 79.81 | 65.18 | 89.92 | 77.40 |
| | x | Word+Syn. | 80.33 | 67.27 | 90.87 | *78.23* |
| (4) Multilayer BiLSTM-CRF | | Word | 76.95 | 63.07 | 89.19 | 74.84 |
| | | Word+Syn. | 77.33 | 64.97 | 91.47 | 75.61 |
| | x | Word | 80.04 | 64.24 | 90.10 | 77.40 |
| | x | Word+Syn. | 80.33 | 67.04 | 90.94 | 78.15 |
| (5) Multilayer BiLSTM-MLP1-CRF | | Word | 76.27 | 60.24 | 90.32 | 73.89 |
| | | Word+Syn. | 77.32 | 62.22 | 91.19 | 75.12 |
| | x | Word | 78.46 | 65.77 | 90.98 | 76.54 |
| | x | Word+Syn. | 78.58 | 65.03 | 92.37 | 76.53 |
| (6) Multilayer BiLSTM-MLP2-CRF | | Word | 77.52 | 62.76 | 90.21 | 75.31 |
| | | Word+Syn. | 77.96 | 63.22 | 91.41 | 75.75 |
| | x | Word | 78.58 | 64.62 | 89.15 | 76.32 |
| | x | Word+Syn. | 80.37 | 67.14 | 91.01 | **78.24** |

All models are trained and evaluated using 10-fold cross validation with the same training sets and test sets. The bold and italic numbers denote the best and the second best results

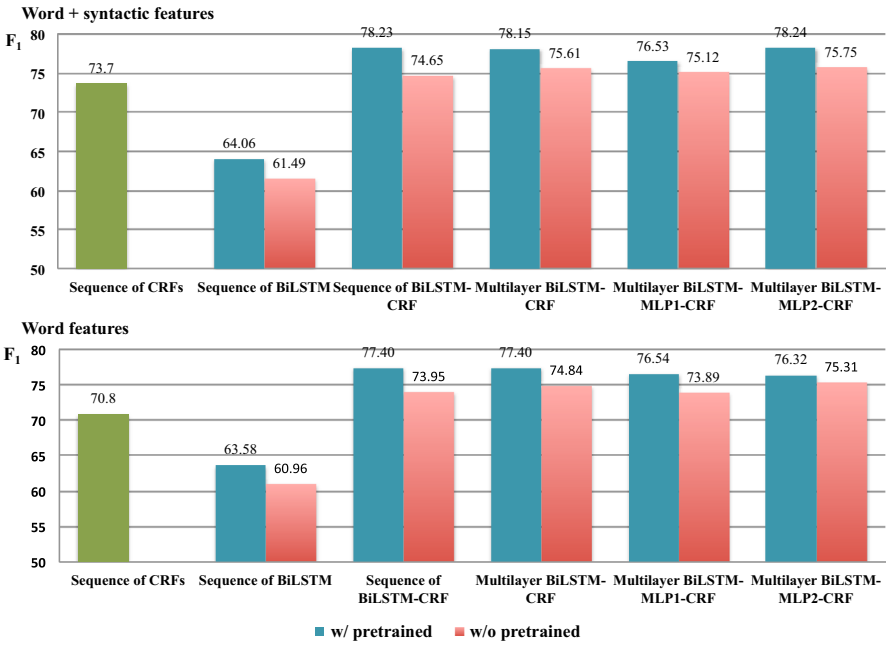


**Fig. 7** Comparison between different models on JCC-RRE dataset

pre-trained embeddings improved the performance from 2 to 4% in $F_1$ scores. Besides, using syntactic features also encourages the models to recognize RE parts better. These features improved the performance from 1 to 2% . Therefore, for each model, using pre-trained embeddings and syntactic features usually produces the best result. Our best model produced an $F_1$ score of 78.24% which outperforms by $\sim$4.5% at $F_1$ score compared to the best baseline.

The experimental results demonstrate that the proposed models can learn implicit features from the annotated corpus. In experiments which do not use any kind of features, the proposed models which only used pre-trained word embeddings achieved significant improvements compared to the sequence of CRFs with a set of syntactic features. For example, without external features, models 3, 4, 5, 6 with pre-trained embeddings produced F1 scores from 76.32 to 77.4% which improved from 3 to 4% compared to 73.7% produced by the baseline CRFs which used syntactic features.

*Comparison between the sequence of BiLSTM-CRFs and Multi-BiLSTM-CRF* The architecture of a BiLSTM-CRF component in the multilayer model are quite similar to a BiLSTM-CRF in the cascading approach. Consequently, the number of parameters, training time, testing time and the performance of these two models are comparable (Table 9). The advantage of the multilayer BiLSTM-CRF compared to the cascading approach is that it is a unified model which simplifies the training and testing process.

**Table 8** Details results on JCC-RRE dataset of all models which used word and syntactic features

| Model | Layer | Tag | P | R | $F_1$ |
|---|---|---|---|---|---|
| (1) Sequence of CRF | 1 | R | 81.13 | 74.92 | 77.90 |
| | | E | 72.95 | 75.86 | 74.38 |
| | 2 | E | 64.63 | 54.49 | 59.13 |
| | 3 | U | 90.91 | 84.94 | 87.82 |
| | *All layers* | *All tags* | 76.05 | 71.49 | 73.70 |
| (2) Sequence of BiLSTM | 1 | R | 61.85 | 74.02 | 67.39 |
| | | E | 52.76 | 69.97 | 60.16 |
| | 2 | E | 55.05 | 59.98 | 57.41 |
| | 3 | U | 82.16 | 85.33 | 83.71 |
| | *All layers* | *All tags* | 58.64 | 70.57 | 64.06 |
| (3) Sequence of BiLSTM-CRF | 1 | R | 82.53 | 80.40 | 81.45 |
| | | E | 78.25 | 78.78 | 78.51 |
| | 2 | E | 69.16 | 65.61 | 67.34 |
| | 3 | U | 91.41 | 90.35 | 90.87 |
| | *All layers* | *All tags* | 79.09 | 77.39 | 78.23 |
| (4) Multilayer BiLSTM-CRF | 1 | R | 83.61 | 79.13 | 81.31 |
| | | E | 77.58 | 80.00 | 78.77 |
| | 2 | E | 67.55 | 66.54 | 67.04 |
| | 3 | U | 90.77 | 91.12 | 90.94 |
| | All layers | All tags | 78.90 | 77.40 | 78.15 |
| (5) Multilayer BiLSTM-MLP1-CRF | 1 | R | 80.32 | 79.01 | 79.66 |
| | | E | 76.11 | 77.56 | 76.83 |
| | 2 | E | 65.77 | 64.30 | 65.03 |
| | 3 | U | 91.32 | 93.44 | 92.37 |
| | All layers | All tags | 76.75 | 76.31 | 76.53 |
| (6) Multilayer BiLSTM-MLP2-CRF | 1 | R | 82.14 | 79.95 | 81.03 |
| | | E | 79.54 | 79.05 | 79.29 |
| | 2 | E | 68.76 | 65.61 | 67.14 |
| | 3 | U | 90.15 | 91.89 | 91.01 |
| | *All layers* | *All tags* | 79.14 | 77.36 | 78.24 |

*Training time, testing time and size of different multilayer models* The size, training time, testing time and performance of different multilayer models are shown in Table 9. In two multilayer models, a Multi-BiLSTM-MLP-CRF possesses several advantages. Firstly, with the same size of hidden layers and input embeddings, a Multi-BiLSTM-MLP-CRF has many fewer parameters than a Multi-BiLSTM-CRF (the size of a Multi-BiLSTM-MLP-CRF is comparable with a single BiLSTM-CRF). Thus, compared to a Multi-BiLSTM-CRF, the training and testing time of Multi-BiLSTM-MLP-CRF is faster. Secondly, although Multi-BiLSTM-MLP-CRF contains fewer parameters, its performance are also competitive with Multi-BiLSTM-CRF.

**Table 9** Number of parameters, training time (per epoch), testing time of all models in JCC-RRE data set

| Model | #params (k) | Training time/epoch (s) | Testing time (1660 sentences) (s) | F1 score |
|---|---|---|---|---|
| Multi-BiLSTM-CRF | 650 | 126 | 48.2 | 78.15 |
| Multi-BiLSTM MLP1-CRF | 213 | 51 | 20.8 | 76.53 |
| Multi-BiLSTM MLP2-CRF | 240 | 55 | 21.2 | 78.24 |
| Sequence of BiLSTM-CRF | 654 | 140 | 52.9 | 78.23 |

Configuration: The size of input word embeddings and the size of hidden layers in MLP and BiLSTM component is 100. All experiments for measuring the time consumption are conducted in the same condition

Figure 8 shows the performance on the development set during the training process of three multilayer models. The trend of three models are the same, they produce the stable results around 100 epochs and the scores do not change much after that. This demonstrated that training a Multi-BiLSTM-MLP-CRF is faster but its performances are equivalent to a Multi-BiLSTM-CRF. In two configurations of Multi-BiLSTM-MLP-CRF, Multi-BiLSTM-MLP2-CRF exhibited a better performance than Multi-BiLSTM-MLP1-CRF (see Table 7). The reason may be that the additional dense layer provides more parameters that help to learn the model better.

*Comparison between the single evaluation and end-to-end evaluation* Table 10 shows the results of the sequence of BiLSTM models on the JCC-RRE dataset in both evaluation methods. The performance of the single evaluation method is usually higher than that of the end-to-end evaluation method. This result is understandable because it showed the dependencies between the tags in a certain layer with the tags of the previous layers. For example, the result for layer 2 with the end-to-end evaluation method is much smaller than the result of this layer with the single layer evaluation method. In the JCC-RRE corpus, if a requisite part and an effectuation part are overlapped, tags represented the requisite part is often located at layer 1 and tags represented the effectuation part is often located at layer 2. Due to these dependencies, the recognition of requisite parts in layer 1 will affect the recognition of effectuation parts in layer 2. On the other hand, if tags of two layers are not related, the result of these two evaluation methods is not greatly different (e.g, layer 3 contains only *Unless* parts and recognizing these parts do not depend on tags of other layers).

### 5.5 Error analysis

Analyzing errors of deep learning systems is more difficult than rule-based systems because the work-flow inside deep learning models is very complicated. We pick some outputs that may express the differences between different configuration. We observed that, in long sentences, syntactic features such as POS tags seem to help to

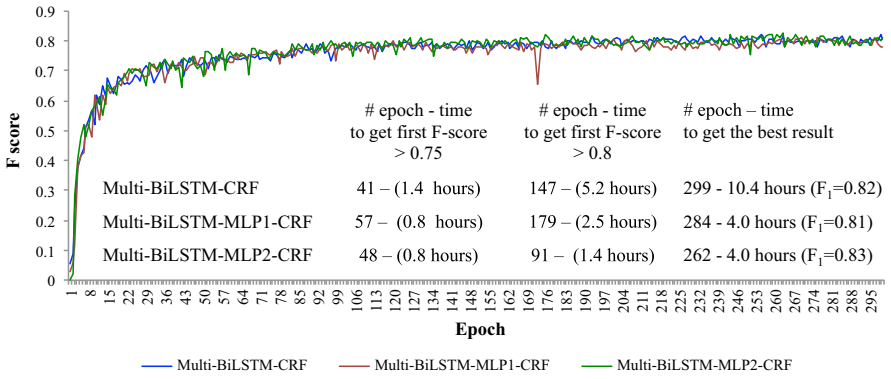


**Fig. 8** Evaluation result on the validation set during the training process of one experiment in the 10-fold cross validation approach of different multilayer models (with syntactic features and pre-trained embeddings

**Table 10** Comparison between end-to-end evaluation and single-evaluation method on JCC-RRE dataset

| Model | Pre. emb. | Features | Layer 1 | Layer 2 | | Layer 3 | |
|---|---|---|---|---|---|---|---|
| | | | | single eval. | end2end eval | single eval. | end2end eval. |
| Sequence of CRFs | | Word | 73.23 | 83.99 | 56.49 | 90.32 | 88.8 |
| | | Word + Syn. | 76.52 | 85.41 | 59.07 | 88.58 | 87.82 |
| Sequence of BiLSTM | | Word | 61.35 | 78.77 | 53.60 | 85.02 | 85.33 |
| | | Word + Syn. | 62.02 | 78.30 | 54.72 | 78.00 | 80.60 |
| | x | Word | 64.17 | 79.78 | 56.56 | 80.80 | 82.71 |
| | x | Word + Syn. | 64.52 | 79.78 | 57.31 | 80.80 | 83.87 |
| Sequence of BiLSTM-CRF | | Word | 76.37 | 85.85 | 61.41 | 91.40 | 88.20 |
| | | Word + Syn. | 77.22 | 86.03 | 61.02 | 91.40 | 90.27 |
| | x | Word | 79.81 | 87.75 | 65.18 | 88.89 | 89.92 |
| | x | Word + Syn. | 80.03 | 88.32 | 67.46 | 91.19 | 90.45 |

predict the boundaries of RE parts better. For example, Table 11 shows outputs of the sequence of BiLSTM-CRF models of two cases: (a) w/o features and (b) w/ features. In case (a), without using syntactic features, the model failed to predict the requisite part *with respect to such portion of that wall that is higher than the lower building*. However, in case (b), the model with syntactic features can predict correctly both two requisite and effectuation parts. If we change the POS of words in the phrase "*with respect to*", the model in case (b) will fail to predict these parts. This points out that not only the phrase "*with respect to*" but also their POS tags are clues that help the model predict RE parts correctly.

In many cases, there is little difference between the output of proposed models and the gold data. However, because we employ the strict matching evaluation, the systems get minus points due to these differences. Table 12 shows an example in

**Table 11** Output of Sequence of BiLSTM-CRF models for the input: "If the height of a wall that separates two neighboring buildings of different heights is higher than the height of the lower building , the preceding paragraph shall likewise apply with respect to such portion of that wall that is higher than the lower building ; ..."

| Word | POS | Gold | w/o features | w/ features | Word | POS | Gold | w/o features | w/ features |
|---|---|---|---|---|---|---|---|---|---|
| If | IN | B-R | B-R | B-R | the | DT | B-E | B-E | B-E |
| the | DT | I-R | I-R | I-R | preceding | VBG | I-E | I-E | I-E |
| height | NN | I-R | I-R | I-R | paragraph | NN | I-E | I-E | I-E |
| of | IN | I-R | I-R | I-R | shall | MD | I-E | I-E | I-E |
| a | DT | I-R | I-R | I-R | likewise | RB | I-E | I-E | I-E |
| wall | NN | I-R | I-R | I-R | apply | VB | I-E | I-E | I-E |
| that | WDT | I-R | I-R | I-R | with | IN | B-R | I-E | B-R |
| separates | VBZ | I-R | I-R | I-R | respect | NN | I-R | I-E | I-R |
| two | CD | I-R | I-R | I-R | to | TO | I-R | I-E | I-R |
| neighboring | JJ | I-R | I-R | I-R | such | JJ | I-R | I-E | I-R |
| buildings | NNS | I-R | I-R | I-R | portion | NN | I-R | I-E | I-R |
| of | IN | I-R | I-R | I-R | of | IN | I-R | I-E | I-R |
| different | JJ | I-R | I-R | I-R | that | DT | I-R | I-E | I-R |
| heights | NNS | I-R | I-R | I-R | wall | NN | I-R | I-E | I-R |
| is | VBZ | I-R | I-R | I-R | that | WDT | I-R | I-E | I-R |
| higher | JJR | I-R | I-R | I-R | is | VBZ | I-R | I-E | I-R |
| than | IN | I-R | I-R | I-R | higher | JJR | I-R | I-E | I-R |
| the | DT | I-R | I-R | I-R | than | IN | I-R | I-E | I-R |
| height | NN | I-R | I-R | I-R | the | DT | I-R | I-E | I-R |
| of | IN | I-R | I-R | I-R | lower | JJR | I-R | I-E | I-R |
| the | DT | I-R | I-R | I-R | building | NN | I-R | I-E | I-R |
| lower | JJR | I-R | I-R | I-R | ; | : | – | – | – |
| building | NN | I-R | I-R | I-R | | | | | |
| , | , | – | – | – | | | | | |

which our system recognizes "*any portion of the gift for which performance has been completed*" as a requisite part while the annotation in gold data is "*portion of the gift for which performance has been completed*". The recognizing of *Unless* parts is very precisely because *Unless* parts are usually presented in specific structures which make them easily recognizable than other parts.

*The effects of sentence length on performance* Table 13 shows the experimental results of multilayer models with different sentence length. The performance on long sentences ( $\geq 90$ words) is low because these sentences are complex and they contain many RE parts in a sentences. However, it is surprising that the performance on short sentences is not high as our expectation. We observed that requisite parts in medium-length sentences usually presented in explicit structures such as "if/in cases" that makes the RE parts in medium-length sentences are easier than short

**Table 12** Output of our the sequence of BiLSTM-CRF models of 3 layers for the input: "; provided , however , that this shall not apply to any portion of the gift for which performance has been completed "

| | | Gold | | | w/o features | | | w/ features | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Layer 1 | Layer 2 | Layer 3 | Layer 1 | Layer 2 | Layer 3 | Layer 1 | Layer 2 | Layer 3 |
| ; | : | – | – | – | – | – | – | – | – | – |
| provided | VBN | – | – | B-U | – | – | B-U | – | – | B-U |
| , | , | – | – | I-U | – | – | I-U | – | – | I-U |
| however | RB | – | – | I-U | – | – | I-U | – | – | I-U |
| , | , | – | – | I-U | – | – | I-U | – | – | I-U |
| that | IN | – | – | I-U | – | – | I-U | – | – | I-U |
| this | DT | – | B-E | I-U | – | B-E | I-U | – | B-E | I-U |
| shall | MD | – | I-E | I-U | – | I-E | I-U | – | I-E | I-U |
| not | RB | – | I-E | I-U | – | I-E | I-U | – | I-E | I-U |
| apply | VB | – | I-E | I-U | – | I-E | I-U | – | I-E | I-U |
| to | TO | – | I-E | I-U | – | I-E | I-U | – | I-E | I-U |
| any | DT | – | I-E | I-U | B-R | I-E | I-U | B-R | I-E | I-U |
| portion | NN | B-R | I-E | I-U | I-R | I-E | I-U | I-R | I-E | I-U |
| of | IN | I-R | I-E | I-U | I-R | I-E | I-U | I-R | – | I-U |
| the | DT | I-R | I-E | I-U | I-R | I-E | I-U | I-R | – | I-U |
| gift | NN | I-R | I-E | I-U | I-R | I-E | I-U | I-R | – | I-U |
| for | IN | I-R | – | I-U | I-R | – | I-U | I-R | – | I-U |
| which | WDT | I-R | – | I-U | I-R | – | I-U | I-R | – | I-U |
| performance | NN | I-R | – | I-U | I-R | – | I-U | I-R | – | I-U |
| has | VBZ | I-R | – | I-U | I-R | – | I-U | I-R | – | I-U |
| been | VBN | I-R | – | I-U | I-R | – | I-U | I-R | – | I-U |
| completed | VBN | I-R | – | I-U | I-R | – | I-U | I-R | – | I-U |
| . | . | – | – | – | – | – | – | – | – | – |

sentences. However, in short sentences, requisite parts usually appear in preposition phrases that may cause some difficulties for recognizing RE parts. Table 14 shows outputs of our systems on short sentences in which the models can fail in several cases.

*The effectiveness of special words on performance* Table 15 shows the evaluation of Multi-BiLSTM-MLP2-CRF on sentences which contains special phrases mentioned in the preceding paragraph including *"if"* and *"in cases"*. It is understandable that the recognition of RE parts in these sentences are more precisely. It achieved an $F_1$ score of 83.29% compared with 78.24% when evaluating on all sentences. However, the recognition of effectuation parts is deficient due to the ambiguity of preposition phrases. For example, for the input sentence *"If there are two or more holders of statutory liens with the same priority*

**Table 13** Experimental results in different sentence length of multilayer models (using features + pre-trained embeddings)

| Length (words) | Multi-BiLSTM-CRF | Multi-BiLSTM-MLP1-CRF | Multi-BiLSTM-MLP2-CRF |
|---|---|---|---|
| < 20 | 71.90 | 71.12 | 72.94 |
| 20-29 | 78.46 | 75.66 | 78.20 |
| 30-39 | 77.38 | 77.89 | 78.58 |
| 40-49 | 82.41 | 81.20 | 82.58 |
| 50-59 | 80.82 | 79.97 | 81.52 |
| 60-69 | 81.67 | 77.71 | 79.31 |
| 70-79 | 78.94 | 75.90 | 80.51 |
| 80-89 | 81.71 | 78.12 | 80.57 |
| >= 90 | 65.33 | 62.92 | 63.86 |
| Overall | 78.15 | 76.53 | 78.24 |

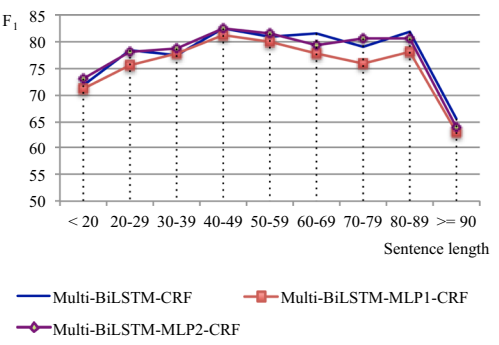


**Table 14** Some sample outputs of our model (Multi-BiLSTM-CRF) on short sentences. In case (a), our model correctly predicts both R and E parts; in case (b), our model predicts the R part correctly, but not the E part; in cases (c) and (d), our model failed to predict both the R and E parts

| a) | Gold | | System | |
|---|---|---|---|---|
| Parties | B-R | B-E | B-R | B-E |
| to | I-R | – | I-R | – |
| an | I-R | – | I-R | – |
| adoption | I-R | – | I-R | – |
| may | – | I-E | – | I-E |
| agree | – | I-E | – | I-E |
| to | – | I-E | – | I-E |
| dissolve | – | I-E | – | I-E |
| the | – | I-E | – | I-E |
| adoptive | – | I-E | – | I-E |
| relationship | – | I-E | – | I-E |
| . | – | – | – | – |

| b) | Gold | | System | |
|---|---|---|---|---|
| A | B-R | B-E | B-R | B-E |
| child | I-R | I-E | I-R | I-E |
| out | I-R | I-E | I-R | – |
| of | I-R | I-E | I-R | – |
| wedlock | I-R | I-E | I-R | – |
| shall | – | I-E | – | I-E |
| take | – | I-E | – | I-E |
| the | – | I-E | – | I-E |
| surname | – | I-E | – | I-E |
| of | – | I-E | – | I-E |
| his/her | – | I-E | – | I-E |
| mother | – | I-E | – | I-E |
| . | – | – | – | – |

| c) | Gold | | System | |
|---|---|---|---|---|
| Neither | B-E | – | B-R | B-E |
| an | I-E | – | I-R | I-E |
| ascendant | I-E | – | I-R | I-E |
| nor | I-E | – | I-R | – |
| a | I-E | – | I-R | – |
| person | I-E | – | I-R | – |
| of | I-E | – | I-R | – |
| greater | I-E | – | I-R | – |
| age | I-E | – | I-R | – |
| may | I-E | – | – | I-E |
| be | I-E | – | – | I-E |
| adopted | I-E | – | – | I-E |
| . | – | – | – | – |

| d) | Gold | | System | |
|---|---|---|---|---|
| The | B-R | B-E | B-E | – |
| benefits | I-R | I-E | I-E | – |
| of | I-R | – | I-E | – |
| the | I-R | – | I-E | – |
| prescription | I-R | – | I-E | – |
| may | – | I-E | I-E | – |
| not | – | I-E | I-E | – |
| be | – | I-E | I-E | – |
| waived | – | I-E | I-E | – |
| in | – | I-E | I-E | – |
| advance | – | I-E | I-E | – |
| . | – | – | – | – |

*with respect to the same object*, *the holders of statutory liens shall be paid in proportion to the amounts of their claims* .", the underlined part is considered as only one effectuation part, but our system recognizes *"the holders of statutory liens"* as a requisite and *"the holders shall be paid in proportion to the amounts of their claims"* as a effectuation parts. Handling these cases can improve the performance of RRE systems.

**Table 15** Evaluation results of Multi-BiLSTM-MLP2-CRF on sentences which contain special phrases such as "if", "in cases". The experimental results of this model on all sentences are shown in Table 8

| Phrase | Layer | Tag | P | R | F |
|---|---|---|---|---|---|
| if | Layer 1 | R | 87.60 | 85.78 | 86.68 |
| | | E | 86.66 | 86.05 | 86.35 |
| | Layer 2 | E | 59.78 | 50.46 | 54.73 |
| | Layer 3 | U | 92.21 | 95.95 | 94.04 |
| | *All layers* | *All tags* | 85.44 | 83.37 | 84.39 |
| in_cases | Layer 1 | R | 86.63 | 86.63 | 86.63 |
| | | E | 85.50 | 83.82 | 84.65 |
| | Layer 2 | E | 55.81 | 52.17 | 53.93 |
| | Layer 3 | U | 94.44 | 91.89 | 93.15 |
| | *All layers* | *All tags* | 84.54 | 83.44 | 83.99 |

## 6 Conclusions

This paper proposes various neural network approaches for recognizing requisite and effectuation parts in legal text. First, we introduced a modification of BiLSTM-CRF that allows one to use external features to recognize non-overlapping RE parts. Then we proposed the sequence of BiLSTM-CRF models and two types of multilayer models to recognize overlapping RE parts including Multi-BiLSTM-CRF and Multi-BiLSTM-MLP-CRF. Our approaches outperform previous approaches significantly and achieve state-of-the-art results on the JPL-RRE dataset. For the JCC-RRE dataset, our approaches outperform CRFs, a strong algorithm for the sequence labeling task. For the recognition of overlapping RE parts, the multilayer models are convenient because they are unified models which simplify the training and testing process but produce competitive results compared to the sequence of BiLSTM-CRF models. In two types of multilayer models, Multi-BiLSTM-MLP-CRF solves limitations of Multi-BiLSTM-CRF because it eliminates redundant components thus the size is smaller, and training time and testing time are faster without diminishing the performance.

## 7 Future work

There are two directions for our future work. Firstly, we can extract new feature sets beside a few syntactic features to improve the performance of RRE systems. The architecture of the model allows us to integrate new features without making any changes. Secondly, we can apply the proposed models to other sequence labeling tasks (e.g. named entity recognition, information extraction, semantic role labeling or discourse parsing) in different domains. In the legal domain, these models can be applied to improve the performance of previous work in legal text analysis such as named entity recognition (Dozier et al. 2010), claims identification (Surdeanu et al. 2010). In addition, these models can also be applied to other domains such as biomedical texts, electronic health-care reports, patent documents, etc.

**Acknowledgements** This work was supported by JSPS KAKENHI Grant Number 15K16048, JSPS KAKENHI Grant Number JP15K12094, and JST CREST Grant Number JPMJCR1513, Japan.

## References

Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166

Boden M (2001) A guide to recurrent neural networks and backpropagation

Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146

Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, Springer, Berlin pp 177–186

Chiu JP, Nichols E (2015) Named entity recognition with bidirectional lstm-cnns. arXiv preprint arXiv:151108308

Dozier C, Kondadadi R, Light M, Vachher A, Veeramachaneni S, Wudali R (2010) Named entity recognition and resolution in legal text. Springer, Berlin, pp 27–43. https://doi.org/10.1007/978-3-642-12837-0_2

Elman JL (1990) Finding structure in time. Cognit Sci 14(2):179–211

Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6645–6649

Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J (2017) LSTM: A search space odyssey. IEEE Trans Neural Netw Learn Syst 28:2222

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:150801991

Karpathy A (2015) The unreasonable effectiveness of recurrent neural networks. Andrej Karpathy blog

Klein D, Manning CD (2003) Accurate unlexicalized parsing. In: Proceedings of the 41st annual meeting on association for computational linguistics. Volume 1, Association for computational linguistics, pp 423–430

Kudo T (2005) CRF++: Yet another CRF toolkit. Software available at https://taku910.github.io/crfpp/

Kudo T, Yamamoto K, Matsumoto Y (2004) Applying conditional random fields to japanese morphological analysis. In: Proceedings of the 2004 conference on empirical methods in natural language processing

Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the eighteenth international conference on machine learning, ICML vol 1, pp 282–289

Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. arXiv preprint arXiv:160301360

Ling W, Chu-Cheng L, Tsvetkov Y, Amir S (2015) Not all contexts are created equal: Better word representations with variable attention

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119

Nakamura M, Nobuoka S, Shimazu A (2007) Towards translation of legal sentences into logical forms. In: Annual conference of the Japanese society for artificial intelligence, Springer, Berlin pp 349–362

Ngo XB, Nguyen LM, Shimazu A (2010) Recognition of requisite part and effectuation part in law sentences. In: Proceedings of (ICCPOL), pp 29–34

Ngo XB, Nguyen LM, Tran TO, Shimazu A (2013) A two-phase framework for learning logical structures of paragraphs in legal articles. ACM Trans Asian Lang Inf Process (TALIP) 12(1):3

Nguyen LM, Bach NX, Shimazu A (2011) Supervised and semi-supervised sequence learning for recognition of requisite part and effectuation part in law sentences. In: Proceedings of the 9th international workshop on finite state methods and natural language processing, association for computational linguistics, pp 21–29

Nguyen TS, Nguyen TD, Ho BQ, Nguyen LM (2015) Recognizing logical parts in vietnamese legal texts using conditional random fields. In: IEEE RIVF international conference on computing & communication technologies-research, innovation, and vision for the future (RIVF), pp 1–6

Nguyen TS, Nguyen LM, Ho BQ, Shimazu A (2016a) Recognizing logical parts in legal texts using neural architectures. In: IEEE eighth international conference on knowledge and systems engineering (KSE), pp 252–257

Nguyen TS, Nguyen LM, Tran XC (2016b) Vietnamese named entity recognition at vlsp 2016 evaluation campaign. In: Proceedings of the fourth international workshop on vietnamese language and speech processing, pp 18–23

Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

Settles B (2004) Biomedical named entity recognition using conditional random fields and rich feature sets. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications, Association for Computational Linguistics, pp 104–107

Sha F, Pereira F (2003) Shallow parsing with conditional random fields. In: Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology-volume 1, Association for computational linguistics, pp 134–141

Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Surdeanu M, Nallapati R, Manning C (2010) Legal claim identification: Information extraction with hierarchically labeled data. In: Proceedings of the 7th international conference on language resources and evaluation

Taku Kudo YM (2002) Japanese dependency analysis using cascaded chunking. In: CoNLL 2002: proceedings of the 6th conference on natural language learning 2002 (COLING 2002 Post-Conference Workshops), pp 63–69

Tanaka K, Kawazoe I, Narita H (1993) Standard structure of legal provisions-for the legal knowledge processing by natural language. Information Processing Society of Japan Natural Language Processing, pp 79–86

Tjong Kim Sang EF, De Meulder F (2003) 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, association for computational linguistics, pp 142–147

Wang P, Qian Y, Soong F, He L, Zhao H (2015a) A unified tagging solution: Bidirectional lstm recurrent neural network with word embedding. arXiv preprint arXiv:151100215

Wang P, Qian Y, Soong FK, He L, Zhao H (2015b) Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. arXiv preprint arXiv:151006168

Zhou J, Xu W (2015) End-to-end learning of semantic role labeling using recurrent neural networks. In: ACL (1), pp 1127–1137