Doctoral Dissertation

Structure Analysis and Textual Entailment Recognition for Legal Texts using Deep Learning

NGUYEN Truong Son

Supervisor: Associate Professor NGUYEN Le Minh

School of Information Science
Japan Advanced Institute of Science and Technology

September, 2018

Abstract

Analyzing the structure of legal documents and recognizing textual entailment in legal texts are essential tasks to understand the meaning of legal documents. They benefit question answering, text summarization, information retrieval and other information systems in the legal domain. For example, recognizing textual entailment is an essential component in a legal question answering system which answers the correctness of user's statements, or a system which checks the contradiction and redundancy of a newly enacted legal article. Analyzing the structure of legal texts has broader applications because it is one of the preliminary and fundamental tasks which support other tasks. It can break down a legal document into small semantic parts so other systems can understand the meaning of the whole legal document easier. An information retrieval system can leverage a structure analysis component to build a better engine by allowing to search on specific regions instead of searching on the whole legal document.

In this dissertation, we study deep learning approaches for analyzing structures and recognizing textual entailment in legal texts. We also leverage the results of the structure analysis task to improve the performance of RTE task. Both of the results are integrated into a demonstrated system which is an end-to-end question answering system which can retrieve relevant articles and answer from a given yes/no question.

In the work on analyzing the structure of legal texts, we address the problem of recognizing requisite and effectuation (RRE) parts because RE parts are special characteristics of legal texts which different from texts in other domains. Firstly, we propose a deep-learning model based on BiLSTM-CRF, which can incorporate engineering features such as Part-of-Speech and other syntactic-based features to recognize non-overlapping RE parts. Secondly, we propose two unified models for recognizing overlapped RE parts including Multilayer-BiLSTM-CRF and Multilayer-BiLSTM-MLP-CRF. The advantages of proposed models are that they possess a convenient design which can train only a unified model to recognize all overlapped RE parts. Besides, it can reduce the redundant parameters, so the training time and testing time are reduced significantly, but the performance is also competitive. We experimented our proposed models on two benchmark datasets including the Japanese National Pension Law RRE and Japanese Civil Code RRE which are written in Japanese and English, respectively. The experimental results demonstrate the advantages of our model. Our model achieves significant improvements compared to previous approaches on the same feature set. Our proposed model and its design can be extended to use other features easily without changing anything.

We then study the deep learning models for recognizing textual entailment (RTE) in legal texts. We encounter the lack of labeled data problem when applying deep learning models. Therefore, we proposed a semi-supervised learning approach with an unsupervised method for data augmentation which is based on syntactic structures and logical structures of legal sentences. The augmented dataset then is combined with the original dataset to train entailment classification models.

RTE in legal texts is also challenging because legal sentences are long and complex. Previous models use the single-sentence approach which considers related articles as a

very long sentence, so it is difficult to identify important parts of legal texts to make the entailment decision. We then propose methods to decompose long sentences in related articles into simple units such as a list of simple sentences, or a list of RE structures and propose a novel deep learning model that can handle multiple sentences instead of single sentences. The proposed approaches achieve significant improvements compared to previous baselines on the COLIEE benchmark datasets.

We finally connect all components of structure analysis and recognizing textual entailment into a demonstration system which is a question answering system that can answer yes/no question in the legal domain on the Japanese Civil Code. Given a statement which a user needs to check whether or not it is correct, the demonstration system will retrieve relevant articles and classify whether the statement is entailed from its relevant articles. Building these systems can help ordinary people and law experts can exploit information in legal documents more effective.

Keywords: Recognizing textual entailment, Natural Language Inference, Legal Text Analysis, Legal Text Processing, Deep learning, Recurrent Neural Network, Recognizing Requisite and Effectuation.

Acknowledgments

First of all, I wish to express my best sincerest gratitude to my principal advisor, Associate Professor Nguyen Le Minh of Japan Advanced Institute of Science and Technology (JAIST), for his constant encouragement, support and kind guidance during my Ph.D. course. He has gently inspired me in researching as well as patiently taught me to be strong and self-confident in my study. Without his consistent support, I could not finish the work in this dissertation

I would like to express the special thanks to Professor Akira Shimazu of JAIST for his fruitful discussions in my research.

I would like to thank Professor Satoshi Tojo, Associate Professor Kiyoaki Shirai of JAIST, and Professor Ken Satoh of National Institute of Informatics for useful discussions and comments on this dissertation.

I would like to thank Associate Professor Ho Bao Quoc from University Of Science, VNU-HCMC for his suggestion and recommendations to study at JAIST.

I am deeply indebted to the Ministry of Education and Training of Vietnam for granting me a scholarship during the three years of my research. Thanks also to "JAIST Research Grant for Students" and JST CREST program for providing me with their travel grants which supported me to attend and present my work at international conferences.

I would like to thank JAIST staff for creating a wonderful environment for both research and life. I would love to devote my sincere thanks and appreciation to all members of Nguyen's laboratory. Being a member of Nguyen's lab and JAIST is a wonderful time of my research life.

Finally, I would like to express my sincere gratitude to my parents, brothers, and sisters for supporting me with great patience and love. I would also like to express my sincere gratitude to my wife. I would never complete this work without her understanding and tolerance. Also, I would like to express my sincere gratitude to my little son. His innocent smiles are the best encouragements to me for completed the dissertation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The legal system of each country is always one of the most important parts which ensures the safety and the development of that country. Law articles in the legal systems must be consistent with other articles. If this requirement is not satisfied, our society will have suffered from the political and social unrest. However, the number of law documents in a legal system is very big so that law experts cannot check the consistency in these documents manually or make mistakes easily. Therefore, it is essential to build knowledge management systems which be able to automatically exam and verify whether a law contains contradictions, whether the law is consistent with related laws, and whether the law has been modified, added, and deleted consistently. Analyzing the structures and recognizing textual entailment in legal texts are two important tasks need to be solved to build these knowledge management systems. These tasks are also important components in question answering, information retrieval, and legal summarization systems which benefit ordinary people and law experts to exploit the information in legal documents more effectively.

Structure analysis in legal texts: Unlike documents such as online news or users comments in social networks, legal texts possess 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]. A logical part is a span of text in a law sentence (clause or phrase) 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., 2010]. The structure of sentences in legal texts is described in detail in Chapter 3. Identify these logical parts in legal sentences is the purpose of the task of requisite-effectuation recognition (called RRE task).

Legal structure analysis such as RRE is a preliminary step to support other tasks in legal text processing such as translating legal articles into logical and formal representations, or building information retrieval, question answering and other supporting systems in legal domain [Nakamura et al., 2007, Katayama, 2007]. For example, in a question

Table 1.1: An example of an application of RRE in a QA system. If the condition of a "**What if**" question matches the requisite part of a sentence, the effectuation part of this sentence is extracted to be an answer

RE analysis	[If *the advertiser offering prizes specifies the period during which the designated act must be performed*]$_{REQUISITE}$, [it shall be presumed that the advertiser has waived its right to revoke.]$_{EFFECTUATION}$
Question	What if *the advertiser offering prizes specifies the period during which the designated act must be performed* ?
Answer	It shall be presumed that the advertiser has waived its right to revoke .

answering (QA) system, if the question in the form of *"What if a CONDITION?"* and if the *REQUISITE* part of a sentence matches the *CONDITION* part of the given question, we can easily conclude that the answer of that question is the *EFFECTUATION* part of that sentence. Table 1.1 shows an example which is an application of RRE in a QA system in the legal domain. In the task of entailment recognition for legal texts, RRE is a step to decompose a long legal sentence into a list of R-E structures that will make the task of entailment recognition become more simple. RRE is also an essential step in a legal paraphrasing system [Shimazu, 2017] which try to rewrite legal paragraphs to increase their readability.

Textual entailment recognition in legal texts: Recognizing textual entailment (RTE) is one of the fundamental tasks in Natural Language Understanding which identifies or classify whether or not the meaning of a text snippet is entailed by the meaning of the second piece of text [Dagan et al., 2006]. This task is a type of natural language inference (NLI) task in which the more relationship between two texts has been explored including *entailment*, *contradiction* and *neutral* [Bowman et al., 2015]). RTE can be important components in many NLP applications such as Question Answering, Information Extraction, Summarization, and Machine Translation Evaluation because these applications need a model to recognize whether or not the meaning of a text is inferred from another.

In legal domain, the task can be seen as checking whether or not a legal statement is entailed from another. RTE is an important component in systems which check whether or not a legal document contains conflicts or redundancies. It also is a core component of a question answering system which answers whether or not a statement is correct. For example, the entailment task in Competition on Information Extraction/Entailment (COLIEE) from 2014 [Kim and Goebel, 2015, Kim et al., 2016c, Kano et al., 2017b] is one kind of RTE in the legal domain which checks whether or not the given question is entailed from its relevant articles. The entailment task in COLIEE is one of two important tasks need to be solved to build an end-to-end question answering systems in the legal domain which can answer Yes/No questions in Japanese Legal Bar exams. Table 1.2 shows an example of the COLIEE entailment task in which a system must give an answer for the question "*The family court may order the commencement of curatorship without the consent of the person in question.*" by finding relevant articles and checking whether the statement is entailed from these articles.

In legal question answering systems, RTE can serve as a ranking model to rank candi-

Table 1.2: An example of RTE in legal texts in the COLIEE dataset

Article	With respect to any person who whose capacity is extremely insufficient to appreciate right or wrong due to any mental disability, the family court may order the commencement of curatorship upon a request by the person in question, his/her spouse, any relative within the fourth degree of kinship, the guardian, the supervisor of the guardian, the assistant, the supervisor of the assistant, or a public prosecutor;provided however, that, this shall not apply to any person in respect of whom a cause set forth in Article 7 exists.
Statement	The family court may order the commencement of curatorship without the consent of the person in question.
Entailment	Yes

dates for a question in the legal domain. If the hypothesis constructed from a candidate is entailed from the passage, the candidate becomes the correct answer. For example, Table 1.3 shows a list of two candidates of a question related to Vietnamese Traffic Law. We can consider the candidate "*from 600,000 VND to 800,000*" to be a correct answer because its corresponding hypothesis "A fine is *from 600,000 VND to 800,000 VND* if an ordinary vehicle with a high beam in the urban area or residential area . " is entailed from the article. However, the remained candidate "*from 300,000 VND to 400,000*" is not a correct answer because its corresponding hypothesis is not entailed from the relevant article.

Table 1.3: RTE as a ranking model to find the answer from a list of candidates. The passage is a snippet of an article in Vietnamese Traffic Law which relevant with the question

Passage	1. A fine of from 300,000 VND to 400,000 VND shall be imposed for one of the following violations: b) Operating the vehicle at a lower speed than that of other vehicles in the same direction without moving to ... 2. A fine of from 600,000 VND to 800,000 VND shall be imposed for one of the following violations: a) Exceeding the speed limits by 5 km/h but less than 10 km/h; b) Honking, revving up the engine, using air horns or high beam in the urban area or residential area, except for emergency vehicles; ...
Question	How much is a fine if an ordinary vehicle with high beam in the urban area or residential area ?
Candidates	a) from 300,000 VND to 400,000 VND b) from 600,000 VND to 800,000 VND
Hypotheses constructed from candidates	a) A fine is from 300,000 VND to 400,000 VND if an ordinary vehicle with high beam in the urban area or residential area b) A fine is from 600,000 VND to 800,000 VND if an ordinary vehicle with high beam in the urban area or residential area

In this thesis, we first study approaches for analyzing components in legal texts and we focus on recognizing requisite and effectuation in legal sentences. We then study methods

for recognizing textual entailment in legal texts. We also apply the results of RRE task to improve the performance for RTE task. Finally, we use these two components into a system which is an end-to-end question answering system which can answer Yes/No question in Japanese Civil Code. These main parts of our thesis are illustrated in Figure 1.1.

1.2 Research Problems and Contributions

Our study focuses on using deep learning methods for legal text analysis and entailment recognition. Deep Learning is a trend of the computer science community in recent years because of its successes in Artificial Intelligent field. Deep learning methods exhibited its extremely successes in many tasks such as speech recognition [Graves et al., 2013], image and video processing [Simonyan and Zisserman, 2014], and Natural Language Processing. In NLP, many powerful deep learning models have been invented for solving a variety of NLP tasks such as machine translation [Bahdanau et al., 2014, Luong et al., 2015], question answering [Sukhbaatar et al., 2015], textual entailment recognition and natural language inference [Parikh et al., 2016, Liu et al., 2016, Rocktäschel et al., 2015, Chen et al., 2016], text categorization [Kim, 2014], Part-of-Speech tagging, Named Entity Recognition, chunking [Lample et al., 2016, Chiu and Nichols, 2015, Wang et al., 2015b, Huang et al., 2015, Wang et al., 2015a, Collobert et al., 2011]. The thesis focus on three main problems as follow:

- **Analyzing structure of legal texts using deep learning:** In this problem, we mainly focus on RRE task. Previous studies only apply conventional algorithms for RRE such as Conditional Random Fields [Ngo et al., 2010, 2013, Nguyen et al., 2011]. We follow the trend of the research community to apply deep learning methods for RRE task. However, current deep learning methods seem to ignore the benefit of engineering features because they have usually experimented on large datasets. Besides, in RRE task, a requisite part and an effectuation part may overlap but there is no unified model to tackle it. Therefore, we address this problem by proposing unified deep learning models for recognizing overlapping RE parts. The contributions of our study in this part are as follows:
  - We propose a deep learning model based on BiLSTM-CRF which allows incorporating external features along with deep learning models.
  - We exploit several features for RRE task including Part-of-Speech and several syntactic-based features.
  - We propose several approaches for recognizing overlapped RE parts including the cascading approach with the use of many BiLSTM-CRF models and the unified model approach.
  - We propose two novel models called Multilayer-BiLSTM-CRF and Multilayer-BiLSTM-MLP-CRF for the unified model approach.

  We experiment our proposed models on two benchmark datasets including the Japanese National Pension Law RRE and Japanese Civil Code RRE datasets which are written in Japanese and English, respectively. The experimental results demonstrate the advantages of BiLSTM-CRF with external features. It achieves significant

improvements compared to previous approaches on the same feature set. Besides, the design of BiLSTM-CRF with external features can be extended to integrate other features easily without changing anything. This proposed model also exhibited significant improvements on Vietnamese Named Entity Recognition.

In recognizing overlapping RE parts, both of the cascading approach and the unified model approach show promising results and can recognize overlapping RE parts. The advantages of proposed models in the unified approach are that they possess a convenient design which can train only a unified model to recognize all overlapped RE parts. In two models of the unified model approach, Multilayer-BiLSTM-MLP-CRF exhibits advantages. It can reduce the redundant parameters. Consequently, the training time and testing time are reduced significantly but the performance is also competitive compared to the cascading approach and Multilayer-BiLSTM-CRF.

- **Recognizing textual entailment in legal texts using deep learning:** In this problem, we first apply several deep learning models for legal entailment task including the sentence-encoding based models and the attention-based models. However, the result is not our expectation because the dataset is too small. Therefore, we then deal with the problem of data augmentation which tries to generate training examples that cover some linguistic phenomena based on the requisite-effectuation and syntactic structures of legal sentences. Then, we combine the generated datasets and the original dataset to train the model for entailment recognition.

  Besides, RTE in legal texts is also challenging because legal sentences are long and complicated. All previous models use the single-sentence approach which considers related articles as a very long sentence. This approach is difficult for models to focus the important parts of articles to make the entailment decision. We then propose methods to decompose long sentences in related articles into simple sentences base on itemization resolution and RE analysis. We then propose a Multi-Sentence entailment model that can handle multiple sentences instead of single sentences. Our proposed approaches exhibited significant improvements compared to previous baselines on COLIEE benchmark datasets.

  The contributions of our study in this part are as follows:

  - We apply several deep learning models for recognizing textual entailment in legal texts.
  - We propose a semi-supervised approach with an unsupervised method for augmentation which is based on the analysis of requisite-effectuation structures and syntactic parse trees of legal sentences.
  - We propose two methods to decompose a long legal sentence into a list of simple sentences such as analyzing itemization expressions and R-E structures of legal sentences.
  - We propose a novel deep learning model that can handle multiple sentences instead of single sentences.

- **Applications to Question Answering systems:** We build a legal question answering system that utilizes all components of structure analysis and recognizing

textual entailment. This system can answer whether or not a statement is correct based on its relevant articles. Given a statement which a user needs to check whether or not it is correct, the system will first retrieve articles in a legal corpus which relevant to the given statement. We use the cosine similarity score to measure the similarity between the question and relevant articles. Besides, we apply n-gram word indexing to improve the performance of the relevant analysis step. The system then classifies whether the statement is entailed from its relevant articles. Our contributions in this study are as follows:

- We propose a method, called n-gram word indexing, which show significant improvement for the information retrieval task. Besides, we also propose a method for query expansion and apply several techniques for data processing.
- We integrate all components in a two-phase question answering system which can answer yes/no questions from users and display into a web interface.

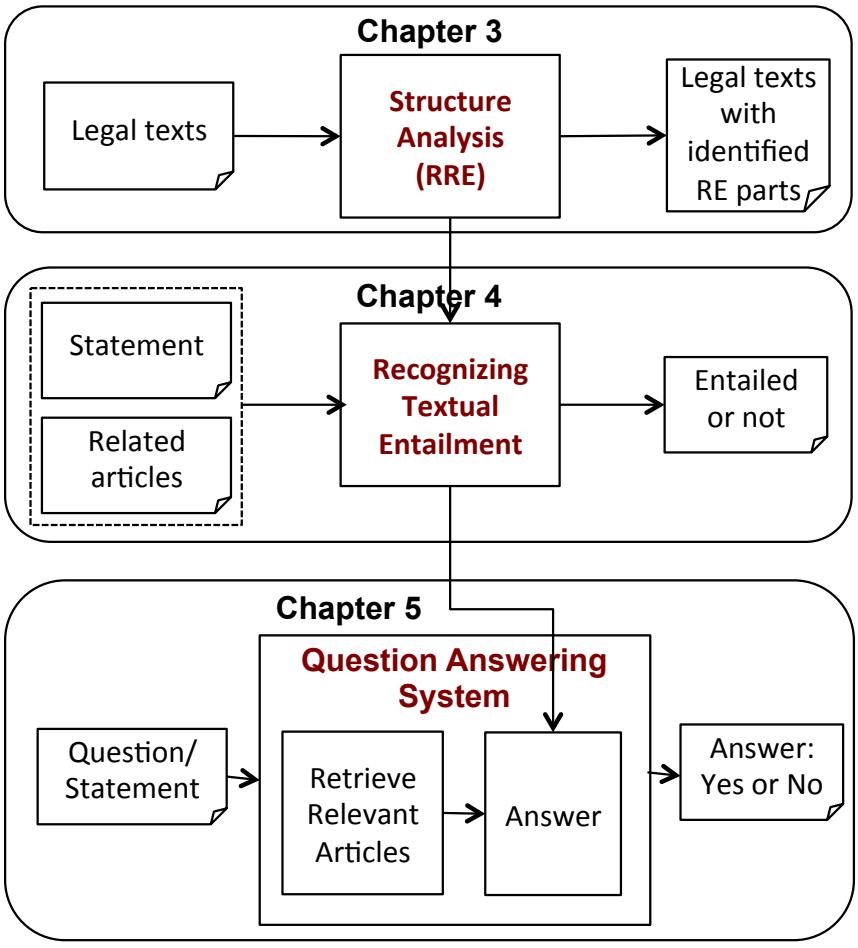


Figure 1.1: Overview of all main parts in our thesis

1.3 Dissertation Outline

The remainder of this dissertation is organized as follows:

Chapter 2 presents the background of learning models for sequence labeling task and recognizing textual entailment. For sequence labeling task, we present the background of Conditional Random Fields and variants of recurrent neural networks for sequence labeling task. For recognizing textual entailment task, we present two typical architectures including sentence encoding-based and attention-based methods.

Chapter 3 addresses the problem of RRE. In this chapter, we first present the structure of legal sentences and the RRE task. We then present the proposed model for recognizing RE parts based on the Long short-term memory that allows integrating engineering features into legal text analysis. However, this model can only recognize non-overlapping RE parts. Therefore, the second part of this chapter presents several proposed methods for recognizing overlapping RE parts by modeling the task as a multilayer sequence labeling task. We first present the cascading approach, which employs a sequence of separate models to recognize RE parts in different layers. This approach is not convenient because it needs to train many single models. We then present a new model, called the multilayer BiLSTM-CRF, to tackle with this inconvenience. However, the multilayer BiLSTM-CRF still contains redundant components and parameters. Therefore, we then present the proposed model, called the multilayer BiLSTM-MLP-CRF, to solve limitations of the multilayer BiLSTM-MLP-CRF. We finally describe experiments and results on Japanese Pension Law RRE and Japanese Civil Code RRE corpus. The feature extraction step for Japanese Civil Code RRE dataset is also described.

Chapter 4 investigates the task of RTE in legal texts. We first describe the COLIEE entailment task. We then present two type of deep learning models for recognizing textual entailment in legal texts including the sentence encoding-based models and the attention-based models. We next present the semi-supervised approach for data augmentation based on syntactic parse trees and requisite-effectuation structures of legal sentences. We then present proposed methods for decomposing a long and complex sentence into a list of simple and short sentences. We next present the proposed model that can handle multiple sentences instead of single sentences. We finally described experiments and results on COLIEE datasets.

Chapter 5 presents applications of recognizing textual entailment and legal structure analysis in a question answering system. We first present the two-phase architecture of the QA system. We then describe components in each phase and how they are connected together. We final present experiments and results each phase in the end-to-end system.

Finally, Chapter 6 presents the summary of our research, some discussions, and future works.

Chapter 2

Background: Learning Methods for Sequence Labeling and Recognizing Textual Entailment

In this chapter, we present a brief introduction of sequence labeling tasks and several supervised learning models for solving this task including Condition Random Fields and Recurrent network-based models (Section 2.1 and 2.1). We then present a brief introduction of recognizing textual entailment and natural language inference and popular deep learning models which is applied to solve these tasks(Section 2.2).

2.1 Learning Methods for Sequence Labeling Task

2.1.1 Sequence Labeling Task

Task definition: Let $\mathbf{x} = \langle x_1, \ldots, x_T \rangle$ be an observation sequence of length T, the task of sequence labeling will assign a sequence of labels $\mathbf{y} = \langle y_1, \ldots, y_T \rangle$ for the input sequence $\mathbf{x}$. Each element x_i is assigned with a label y_i where y_i is a categorical value which belongs to a label set $\mathbb{C}$.

Many tasks in Natural Language Processing can be formulated as a sequence labeling problem such Part-of-Speech (POS) tagging, shallow parsing (chunking), Named entity recognition (NER) (see Table 2.1). Given a sentence as a sequence of words, POS tagging task will assign a single POS label for each word in the input sequence. In other tagging tasks such as NER or chunking, an entity or a phrase may consist of more than one word, IOB tagging scheme is usually used to mark the boundary of the entity or the phrase. For example, words "New" and "York" in Table 2.1 are assigned labels B-LOC and I-LOC to mark the entity "New York". In IOB tagging scheme, words which belong to the beginning or the inside part of an entity are assigned with *B-* and *I-* tags, and *O* tags are used for words that do not belong to any entity.

In the study in Chapter 3, we also formulate RRE task as a sequence labeling task which tries to assign labels to words or phrases to mark the boundary of requisite or effectuation parts. Table 2.2 shows RRE as a sequence labeling problem which will assign B-A, I-A for antecedent parts and B-C, I-C for consequent parts. This task is presented in detail in Chapter 3.

A sequence labeling problem can be solved using different techniques, but supervised

Table 2.1: POS, Chunking and NER as sequence labeling problems

Input	x_1 John	x_2 lives	x_3 in	x_4 New	x_5 York	x_6 and	x_7 works	x_8 for	x_9 the	x_{10} European	x_{11} Union	x_{12} .
Output	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}
POS	NNP	VBZ	IN	NNP	NNP	CC	VBZ	IN	DT	NNP	NNP	.
Chunking	B-NP	B-VP	B-PP	B-NP	I-NP	O	B-VP	B-PP	B-NP	I-NP	I-NP	O
NER	B-PER	O	O	B-LOC	I-LOC	O	O	O	O	B-ORG	I-ORG	O

Table 2.2: RRE as a sequence labeling problem

Input	被保険者期間を	計算する	場合には、	月による	ものと	する。
Output	**B-A**	**I-A**	**I-A**	**B-C**	**I-C**	**I-C**
A:被保険者期間を計算する場合には、(When a period of an insured is calculated,) **C**:月によるものとする。(it is based on a month.)						

learning methods are preferred. We will describe several learning methods for sequence labeling tasks in the next section.

2.1.2 Conditional Random Fields

Conditional Random Fields (CRFs) are probabilistic models that are used to segment and label sequential data. CRFs got the lower error rate than other probabilistic models such as Hidden Markov Model (HMM) or Maximum Entropy Markov Model (MEMM) [Lafferty et al., 2001]. Given an input sequence $\mathbf{x}$, CRFs will define the probability of a label sequence $\mathbf{y}$ given the input sequence $\mathbf{x}$ as a normalized product of potential functions [Leaman and Gonzalez, 2008]. Each potential function has the form of:

$$\mathbf{exp}(\sum_j \lambda_j t_j(y_{i-1}, y_i, \mathbf{x}, i) + \sum_k \mu_k s_k(y_i, \mathbf{x}, i)) \tag{2.1}$$

where $t_j(y_{i-1}, y_i, \mathbf{x}, i)$ is a transition feature function of the entire observation sequence $\mathbf{x}$ and the labels at positions i and $i-1$ in the label sequence; $s_k(y_i, \mathbf{x}, i)$ is a state feature function of the label at position i and the observation sequence; and λ_j and μ_k are parameters to be estimated from training data. The probability of a label sequence $\mathbf{y}$ given an observation sequence $\mathbf{x}$ then can be written as follows:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \lambda, \mu) &\propto \prod_i \mathbf{exp}(\sum_j \lambda_j t_j(y_{i-1}, y_i, \mathbf{x}, i) + \sum_k \mu_k s_k(y_i, \mathbf{x}, i)) \\ &= \mathbf{exp}(\sum_i \sum_j \lambda_j t_j(y_{i-1}, y_i, \mathbf{x}, i) + \sum_i \sum_k \mu_k s_k(y_i, \mathbf{x}, i)) \end{aligned}$$

Training CRFs is the process to estimate the value of λ and μ to maximize the likelihood function with respect to the training data which can be done using gradient descent. After parameters are estimated, the inference in CRFs is the process of searching the output label sequence which has the highest probability for an input observation sequence. The inference process can be done by using dynamic programming algorithms such as Viterbi [Forney, 1973].

CRFs approaches are used successfully in many task such as morphological analysis [Kudo et al., 2004], NER in biomedical and chemical documents [Settles, 2004, Rocktäschel et al., 2012], recognizing logical parts in legal texts [Ngo et al., 2010, Nguyen et al., 2015], information extraction in academic papers [Peng and McCallum, 2006], shallow parsing [Sha and Pereira, 2003]. However, the development of deep learning research with many powerful models provides better solutions for sequence labeling problems. We will present the background of deep learning models for sequence labeling problems in next sections.

2.1.3 Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM)

A recurrent neural network a kind of neural networks that can operate on sequential data, which is suitable for solving sequence labeling tasks.

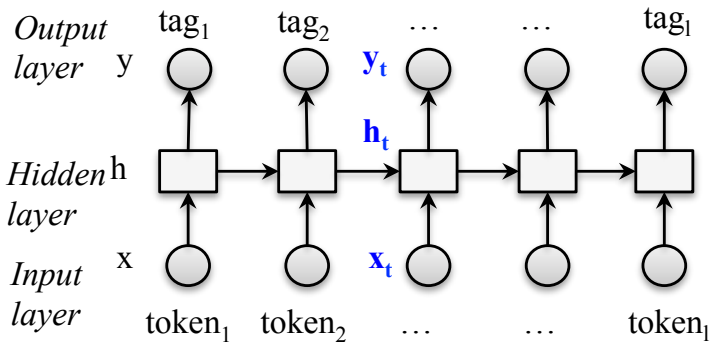


Figure 2.1: Recurrent neural networks

Figure 2.1 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, $\mathbf{x} = (x_1, x_2, ..., x_l)$ represents input embedding vectors of a sequence of tokens and $\mathbf{y} = (y_1, y_2, ..., y_l)$ 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 (token$_t$). 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.2.2). Each $\mathbf{v}_t \in \mathbb{R}^T$, which represents the probability distribution over tags of token$_t$, will then be computed from $\mathbf{h}_t$ using the *softmax* activation function (Eq.2.3). Finally, the output tag $y_t \in [1, T]$ is obtained using *argmax* (Eq.2.4). The values of the hidden and output layers of an RNN are computed as follows:

$$\mathbf{h}_t = f(\mathbf{U}x_t + \mathbf{W}\mathbf{h}_{t-1}) \tag{2.2}$$

$$\mathbf{v}_t = g(\mathbf{V}\mathbf{h}_t) \tag{2.3}$$

$$y_t = \arg\max_{i \in [1,T]} \mathbf{v}_{t_i} \tag{2.4}$$

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:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2.5}$$

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

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. Long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997], a variant of RNNs, solves this limitation by incorporating a memory cell that can 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:

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

where σ is the logistic sigmoid function, and $\odot$ is the element-wise product. $\mathbf{i}$, $\mathbf{f}$, $\mathbf{o}$ and $\mathbf{c}$ 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.

$$loss = -\frac{1}{l}\sum_{t=1}^{l}\sum_{i=1}^{T} \text{ygold}_{t_i} \log(\mathbf{v}_{t_i}) \tag{2.6}$$

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

2.1.4 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 2.2 shows the architecture of a BiLSTM in which the hidden state $\mathbf{h}_t$ represented for knowledge learned from the token_t and its context is the concatenation of forward and backward hidden states.

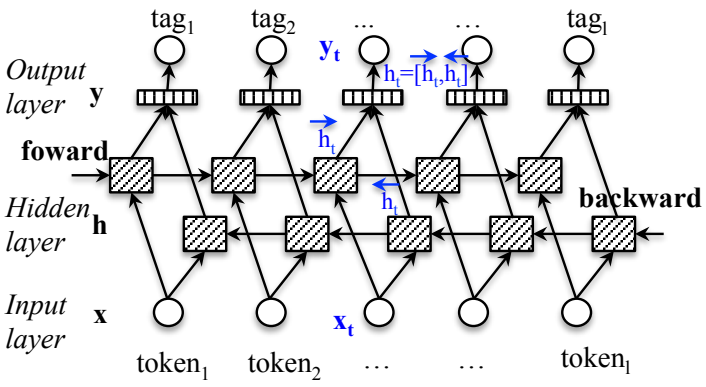


Figure 2.2: Bidirectional Long short-term memory model

2.1.5 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.

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,b].

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:

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{l} A_{y_i, y_{i+1}} + \sum_{i=1}^{l} P_{i, y_i} \tag{2.7}$$

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*:

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X}, \mathbf{y})}}{\sum_{\tilde{y} \in \mathbf{Y_X}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})}} \tag{2.8}$$

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

$$\begin{aligned} \log(p(\mathbf{y}|\mathbf{X})) &= s(\mathbf{X}, \mathbf{y}) - \log(\sum_{\tilde{y} \in \mathbf{Y_X}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})}) \\ &= s(\mathbf{X}, \mathbf{y}) - \operatorname*{logadd}_{\tilde{y} \in \mathbf{Y_X}}(s(\mathbf{X}, \tilde{\mathbf{y}})) \end{aligned} \tag{2.9}$$

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:

$$\mathbf{y}^* = \arg\max_{\tilde{y} \in \mathbf{Y_X}} s(\mathbf{X}, \tilde{\mathbf{y}}) \tag{2.10}$$

where Equation 2.9 and 2.10 can be calculated using a dynamic programming algorithm (e.g Viterbi).

2.1.6 The Effectiveness of BiLSTM-CRF

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. Besides, 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, in Chapter 3, we will present our proposed models which are based on BiLSTM-CRF to deal with the RRE task in legal texts. Firstly, we proposed the single BiLSTM-CRF with features to recognize non-overlapping RE parts. Secondly, 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.

2.2 Deep Learning Models for Recogizing Textual Entailment

2.2.1 Recognizing Textual Entailment (RTE)

Task definition: Recognizing Textual Entailment (RTE) proposed in [Dagan et al., 2006, 2013] is a fundamental task in Natural Language Understanding . The task of RTE is to decide whether a the meaning of a text (H: hypothesis) can be inferred (or entailed) from the meaning of another text (T: Text). The entailment relationship between T and H is a directional relationship. Table 2.3 shows examples of RTE task.

Textual entailment is one type of natural language inference (NLI). In NLI, the semantic relationship between two texts could be a contradiction, neural beside entailment. Compared to other tasks in NLP (e.g., part-of-speech tagging, named entity recognition), NLI is one of the difficult tasks because the relationship between two texts depends not only on the surface the texts but also the meaning of the texts which is difficult to identify.

Text	Hypothesis	Judgment
Norways most famous painting, 'The Scream' by Edvard Munch, was recovered Saturday, almost three months after it was stolen from an Oslo museum.	Edvard Munch painted 'The Scream'.	YES
Arabic, for example, is used densely across North Africa and from the Eastern Mediterranean to the Philippines, as the key language of the Arab world and the primary vehicle of Islam	Arabic is the primary language of the Philippines.	NO

Table 2.3: Examples of RTE task

Text	Hypothesis	Judgment
A man inspects the uniform of a figure in some East Asian country.	The man is sleeping.	contradiction
An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor.	neutral
A soccer game with multiple males playing.	Some men are playing a sport.	entailment

Table 2.4: Examples of natural language inference

Applications of RTE: An natural language understanding component such as RTE is essential for many NLP applications such as information retrieval, question answering, automatic summarization.

In open-domain question answering, an RTE engine is a key component to rank candidates. A candidate answer should be considered correct if and only if the corresponding hypothesis is entailed by the candidate passage from which the candidate was extracted [Dagan et al., 2013]. For example, consider the question "Who painted 'The Scream'?". After the relevant passage "Norways most famous painting, 'The Scream' by Edvard Munch, was recovered Saturday ..." was retrieved and analyzed, 'Edvard Munch' is a candidate answer. We can conclude that 'Edvard Munch' is a correct answer because the corresponding hypothesis "Edvard Munch painted 'The Scream'." is entailed from the relevant passage (see the first example in Table 2.3). In legal domain, people usually want to ask whether a statement is correct, an RTE component for the legal domain is the key component in such question answering systems. Given a statement, if the relevant article of the statement, which can be obtained by the retrieval phase, entails that statement, we can conclude that the statement is *True*, otherwise *False*.

Approaches for RTE and NLI: Early work on natural language inference and recognizing textual entailment textual entailment as been performed on rather small datasets (e.g. RTE-1 to RTE-7 [Bar Haim et al., 2006, Dagan et al., 2006, 2010, Bentivogli et al.]) with more conventional methods. In that time, supervised learning methods (such as SVM, decision tree, Ada-boost) with a set of engineering features are usually used for this task [Malakasiotis and Androutsopoulos, 2007, Zanzotto et al., 2009, Gaona et al., 2010]. Recently, with the availability of large annotated datasets such as SNLI [1] [Bowman et al., 2015], many deep learning models have been invented for NLI which exhibit signif-

[1] https://nlp.stanford.edu/projects/snli/

icant improvements compared to conventional method. Besides, applying deep learning to problems in the legal domain is quite new for the research community. Due to those reasons, we want to apply and propose deep learning models for RTE in legal texts. Several popular and state-of-the-art deep learning models for RTE and NLI are presented in the next sections.

2.2.2 Deep Learning Approaches for RTE and NLI

The basic idea: Given an *text* and a *hypothesis*, a deep learning model for NLI is a complex function $\mathcal{M}$ (also called model $\mathcal{M}$) which will compute the output from the input text and hypothesis.

$$y = \mathcal{M}(t, h, \theta) \tag{2.11}$$

where t, h are the numerical representations of the text and the hypothesis; θ is all parameters of model $\mathcal{M}$ which will be estimated during the training process based on an annotated corpus; y is a vector which represented for the probability distribution over expected classes. For example, in the task of NLI with three classes (entailment, contradiction, neural), y is a vector with a length of 3;

$\mathcal{M}$ is a complex function may consist of many sub-functions which each can be considered as a layer. Thus, the input is passed through many layers. The output of a layer is the input of other layers. Finally, the final layer will compute the output y. The final layer usually uses the $softmax$ function to produce the probability distribution vector.

Sentence encoding-based methods and attention-based methods are two popular methods which are used for natural language inference task.

Sentence encoding-based methods

Figure 2.3 shows the general architecture of a sentence encoding-based method for NLI. The main idea of this method is that it encodes **t** and **h** into two vectors independently by using an encoding layer. These two obtained vectors then are concatenated, the vector then will be transformed through several layers before passing to the final layer to make the classification decision. In sentence encoding-based methods, there is not an explicit comparison between an element in t and an element in h.

Below are important steps in a sentence encoding-based method:

- **Word representation:** in this step, words in t and h are presented by d-fixed length vectors (call embeddings). Text and hypothesis then are represented by two vector sequences $t = \langle a_1, a_2, \ldots, a_n \rangle$ and $h = \langle b_1, b_2, \ldots, b_m \rangle$. These vectors can be initialized randomly or from pre-trained embedding sources.

- **Sentence encoding:** This step uses a method to encode a sentence (a sequence of vectors) into a single vector. There are different methods. For example, we can use a simple method such as summation all words embedding vectors in the input sentence (CBOW). Besides, we can use other neural networks such as convolutional neural networks, vanilla recurrent neural network or its variants such as long short-term memory networks or gated recurrent units. After text and hypothesis are encoded into two single vectors, they then are combined into a single vector. Concatenation of vectors is usually used for the combination step. If t and h can be presented as

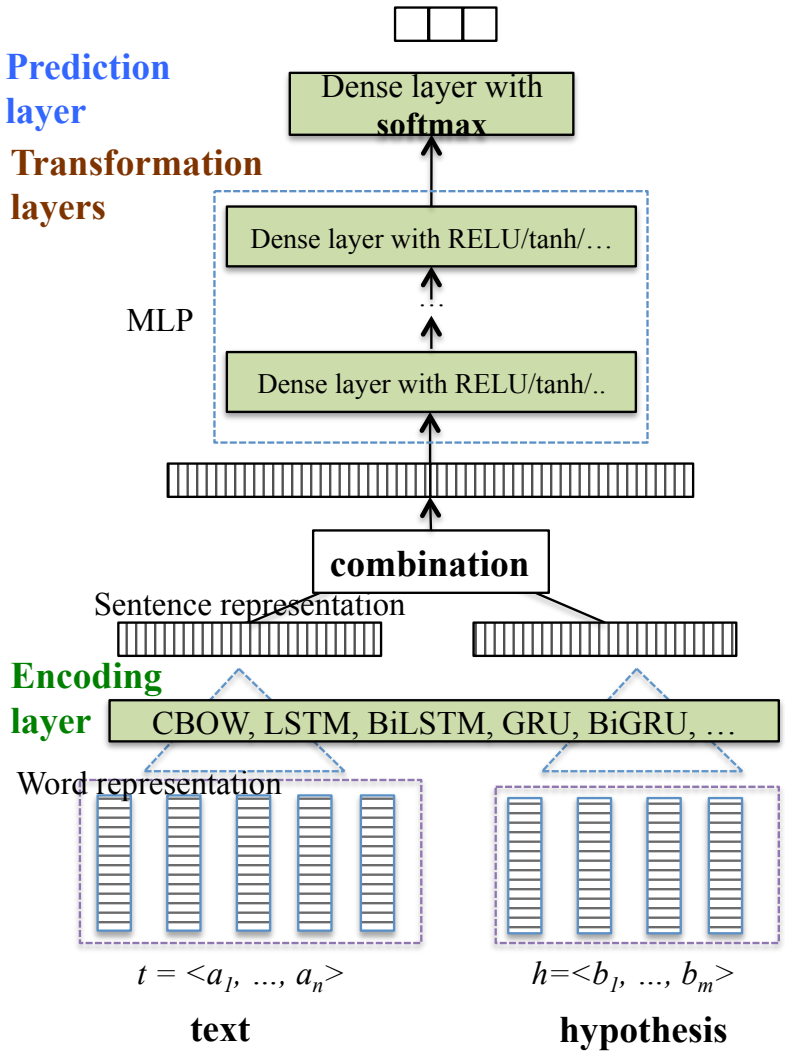


Figure 2.3: A general architecture of sentence encoding-based methods

trees (e.g., syntactic parse trees or dependency trees), a tree encoder may be used to encode these sentences.

- **Transformation:** This step use several fully connected layers to transform the combined vector. Different activation functions can be used in these layers can be used such as *RELU*, *sigmoid*, *tanh*. Besides, other techniques of deep learning field can be used such as dropout [Srivastava et al., 2014], batch normalization [Ioffe and Szegedy, 2015].

- **Prediction:** The output of the transformation step will be passed into the final layer to make the classification decision. The final layer usually uses a $softmax$ activation function to produce a probability distribution vector over expected classes.

This only is a general architecture, when adopting a sentence encoding-based model into a specific task, there are many things need to be tuned such as the size of word embeddings, the encoding method, the size of sentence vector representation, the number of fully connected layer and the type of activation functions, etc.

Model	#Para.	Train	Test
(1) Handcrafted features [Bowman et al., 2015]	-	99.7	78.2
(2) 100D LSTM encoders [Bowman et al., 2015]	221k	84.8	77.6
(3) 300D LSTM encoders [Bowman et al., 2016]	3.0M	83.9	80.6
(4) 1024D pretrained GRU encoders [Vendrov et al., 2015]	15M	98.8	81.4
(5) 300D tree-based CNN encoders [Mou et al., 2016]	3.5M	83.3	82.1
(6) 300D SPINN-PI encoders [Bowman et al., 2016]	3.7M	89.2	83.2
(7) 600D BiLSTM intra-attention encoders [Liu et al., 2016]	2.8M	84.5	84.2
(8) 300D NSE encoders [Munkhdalai and Yu, 2016a]	3.0M	86.2	84.6
(9) 100D LSTM with attention [Rocktäschel et al., 2015]	250K	85.3	83.5
(10) 300D mLSTM [Wang and Jiang, 2015]	1.9M	92	86.1
(11) 450D LSTMN with deep attention fusion [Cheng et al., 2016]	3.4M	88.5	86.3
(12) 200D decomposable attention model [Parikh et al., 2016]	380K	89.5	86.3
(13) 11) + Intra-sentence attention [Parikh et al., 2016]	580K	90.5	86.8
(14) 300D NTI-SLSTM-LSTM [Munkhdalai and Yu, 2016b]	3.2M	88.5	87.3
(15) 300D re-read LSTM [Sha et al., 2016]	2.0M	90.7	87.5
(16) 300D btree-LSTM encoders Paria et al. [2016]	2.0M	88.6	87.6
(17) 600D ESIM [Chen et al., 2016]	4.3M	92.6	88
(18) HIM (600D ESIM + 300D Syntactic tree-LSTM) [Chen et al., 2016]	7.7M	93.5	88.6

Table 2.5: Performance of different inference models for NLI

Many research apply sentence encoding-based methods for NLI due to the its simple architecture (e.g. models 2-8 in Table 2.5). The model in [Bowman et al., 2015] encodes the text and the hypothesis into two 100-dimensional vectors using LSTM or CBOW. It also uses a 3-layer MLP with *tanh* activation function to transform the concatenated vector before passing it to the final layer. The model in [Bowman et al., 2016] and [Mou et al., 2016] encodes the text and using a tree-based encoders. Liu et al. [2016] employed a BiLSTM to encode sentences and model the relationship between words in a sentence using the inner-attention technique.

Attention-based methods

Attention-based methods show advantages in comparison with sentence encoding-based methods because it can attend the semantic information between some parts in the text and the hypothesis. The comparison between *text* and *hypothesis* in sentence encoding-based models are conducted at the sentence level. However, in attention-based models, they are compared at different levels: sentence level, phrase level or word level. An attention-based model can use methods in sentence encoding-based methods to encode sentences. However, we use the term "Sentence encoding-based models" to mention models which only use sentence encoders but the attention mechanism.

There are many variants of attention-based models for NLI tasks (e.g. models 9-18 in Table 2.5). The model in [Rocktäschel et al., 2015] uses two LSTMs and conditional encoding to encode text t and hypothesis h in to hidden states. It then computes word-by-word attention of words in t and h based on those hidden states. Later, Wang and Jiang [2015] improved this model by enforcing word-by-word matching explicitly. Parikh et al. [2016] proposed a simple but very effective decomposable attention model. The model decomposes the NLI problem into sub-problems in which every word pairs between text t and hypothesis h are compared then aggregated before making entailment decision.

Chen et al. [2016] proposed a model for combining sequential and tree representations

for natural language inference. The model, called ESIM, first uses LSTM to encode the text and the hypothesis into a list of hidden states. The attention between h and t then is computed based on the hidden states. Besides, they also employed a Tree-LSTM to obtained the enhanced representations of text and hypothesis based on their parse trees.

In Chapter 4, we present in detail several sentence encoding-based and attention-based models which we will apply for recognizing textual entailment in legal texts. We choose both of basic models and state-of-the-art models for our experiments.

2.3 Training deep learning models

Training neural networks A neural network can be trained by 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] or other variants such as Adam optimizer [Kingma and Ba, 2014], Adadelta optimizer [Zeiler, 2012]. Besides, 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.

Chapter 3

RRE in Legal Texts as Single and Multiple Layer Sequence Labeling Tasks

This chapter presents our study for recognizing requisite and effectuation (RE) parts in Legal Texts. Firstly, we give an introduction to the RRE task and the motivation of using deep learning models for this work (Section 3.1). We then present the structure of legal sentence and introduce the RRE task and present the RRE task as a sequence learning task (Section 3.2). Section 3.3 describes our proposed models for recognizing non-overlapping and overlapping requisite and effectuation parts. Section 3.4 describes our experiments including datasets, experimental settings, results and some discussions. Finally, our conclusions and future work are described in Section 3.5.

3.1 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 Section 3.2.

Recognizing requisite and effectuation parts in legal texts 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 some 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. Besides, 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].

In this study, we propose 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 reduced significantly but the performance is still competitive.

3.2 RRE Task

3.2.1 Structure of Legal Sentences

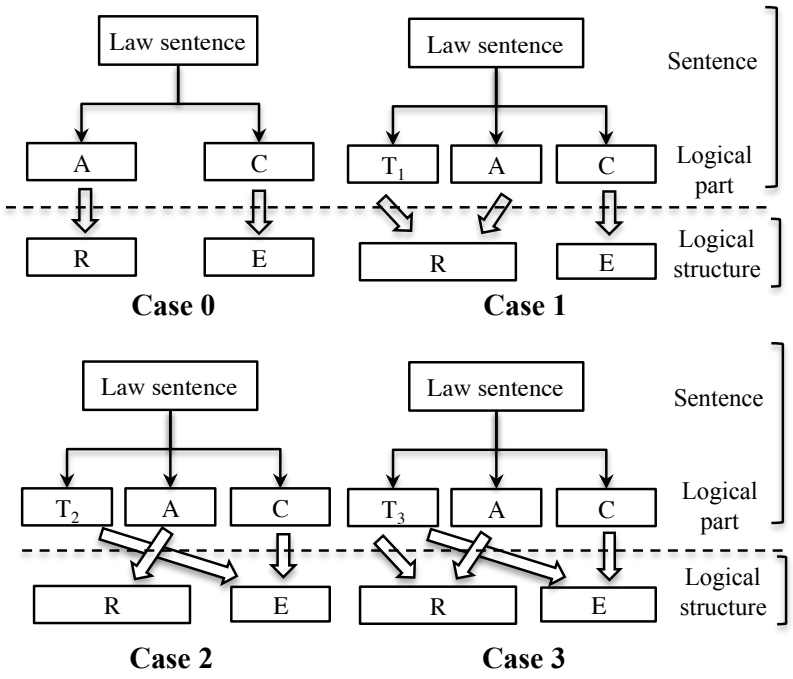


Figure 3.1: Four cases of the logical structure of a law sentence. A represents a antecedent part, C represents a consequent part, T_i represents a topic part [Ngo et al., 2010].

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:

$$\text{requisite part} \Rightarrow \text{effectuation part.}$$

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 Figure 3.1 [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 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]. Table 3.1 and 3.2 show some examples in our experimental datasets.

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 3.2). 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 3.1, 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-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.

Table 3.1: 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 paren1ts while they are married **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 **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 **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 **E:** his/her act may not be rescinded *(non-overlapped): Case 0*

Table 3.2: 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> <A> When a period of an insured is calculated, </A> <C> it is based on a month. </C>	R: A 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> <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>	R: T1 & A 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> <A> If grounds for suspending payment have arisen<A> <T2>insurance benefits in pension form</T2> <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 E: T2 & C

3.2.2 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 $s = \{w_1, w_2, ..., w_l\}$ 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 3.3 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 Section 3.3.

Table 3.3: IOB notation in single and multiple layer RRE dataset. 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.

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

3.3 Proposed Models

3.3.1 The Single BiLSTM-CRF with Features to Recognize Non-overlapping RE Parts

Figure 3.2 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. 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*" (Figure 3.2) 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.

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

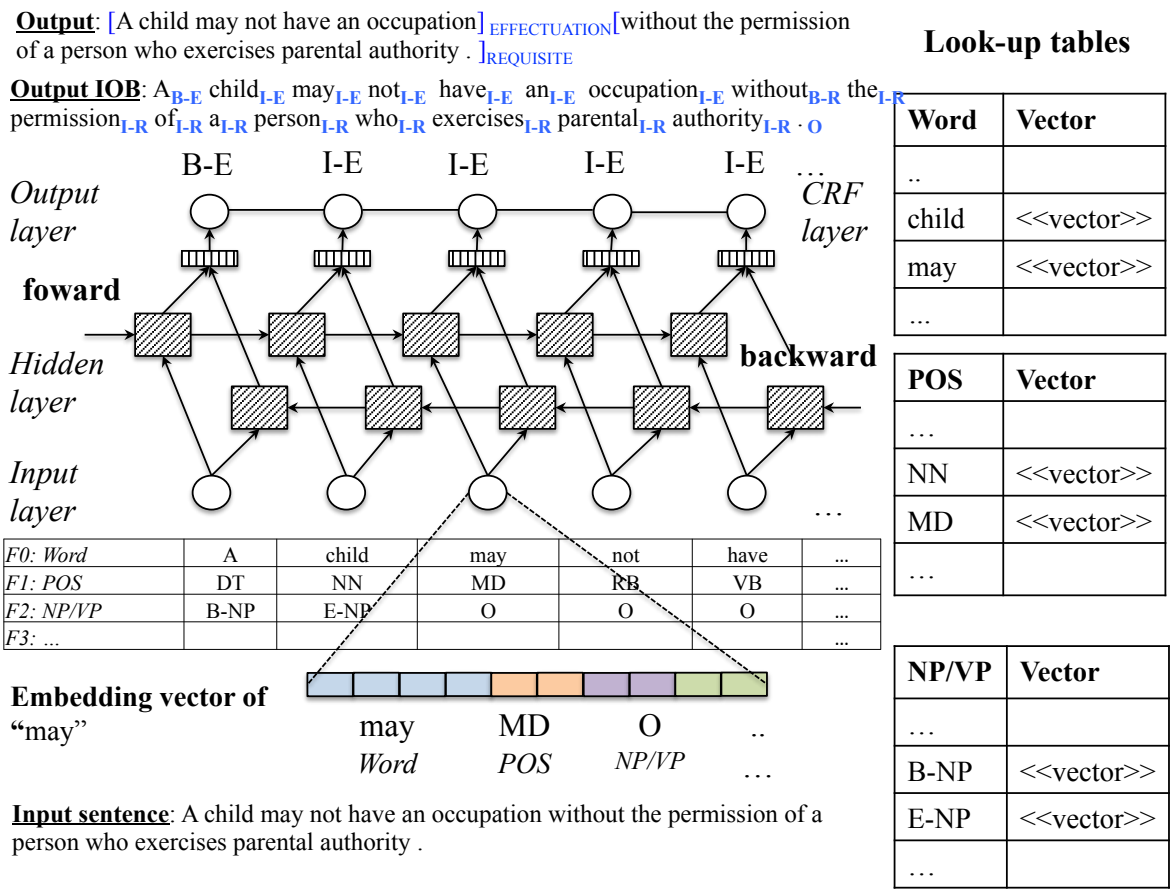


Figure 3.2: BiLSTM-CRF with features to recognize non-overlapping RE parts

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 (Figure 3.2).

If the CRF layer is used, we use the negative of log-probability (Eq. 2.9) as the loss function. Otherwise, the cross-entropy loss (Eq. 2.6) 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 3.3.2. 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 of BI-LSTM-CRF with features

```
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
```

3.3.2 The Cascading Approach to Recognize Overlapping RE Parts

Recognizing overlapping RE parts can be viewed as a multilayer sequence labeling task mentioned in section 3.2. 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 3.3). 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 section 3.3.1 to recognize RE parts in all layers. Figure 3.3 illustrates the cascading approach and 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.

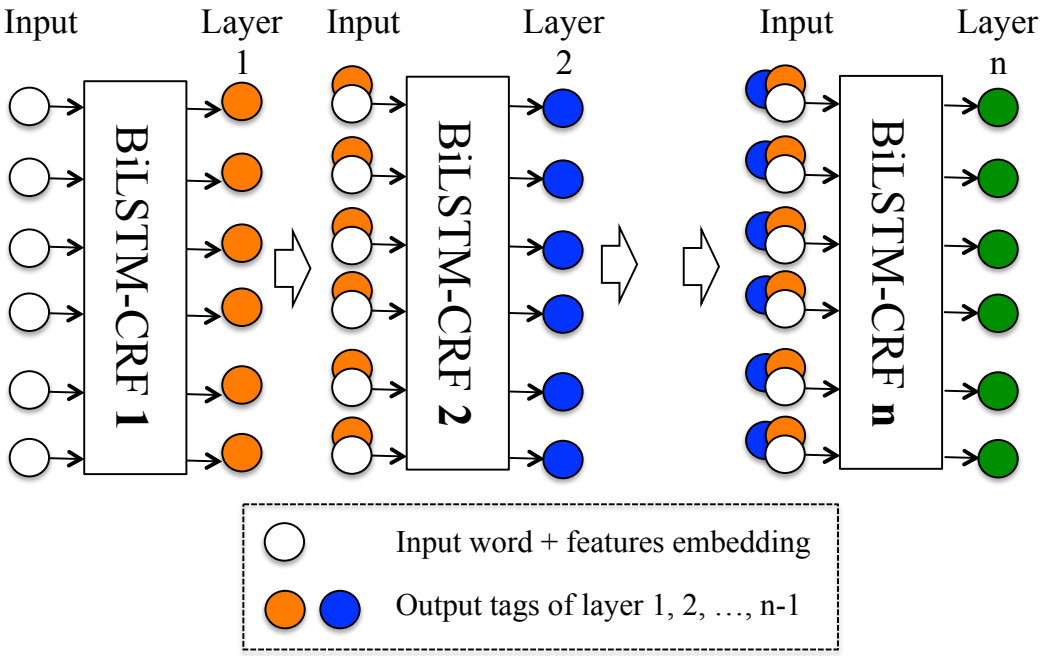


Figure 3.3: The cascading approach for recognizing overlapping RE parts.

3.3.3 Multi-BiLSTM-CRF to Recognize Overlapping RE Parts

The use of n separate models in the cascading approach to recognize overlapping RE parts is inconvenient for training and prediction because we must train n models separately 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 Figure 3.4.

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

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

procedure TRAINSEQUENCE(Corpus, featureTypes)
  $n \leftarrow$ number of layer in the training corpus
  for $i \in 1..n$ **do** ▷ Train a single BiLSTM-CRF model m_i which is responsible to predict the tag at layer i^{th}
    if $i = 1$ **then**
      trainSingle(Corpus, featureTypes, None) ▷
    else
      tags $\leftarrow$ *tagsOfLayers*(Corpus, $[1, i-1]$)
      trainSingle(Corpus, featureTypes, tags) ▷ Using tags in layers 1 to $i-1$ as features to train the model i^{th}
    end if
  end for
end procedure
procedure PREDICTSEQUENCE(test, models)
  outputTagsOfAllLayers $\leftarrow \emptyset$
  $n \leftarrow$ number of layer in the training corpus
  tagsOfPreviousLayers $\leftarrow$ None
  for $i \in 1..n$ **do** ▷ Use model m_i and tags of layers 1 to $i-1$ to predict tag at layer i
    tags $\leftarrow$ *predictSingle*(test, models[i], tagsOfPreviousLayer)
    outputTagsOfAllLayers $\leftarrow$ outputTagsOfAllLayers $\cup$ tags
    tagsOfPreviousLayers $\leftarrow$ tagsOfPreviousLayers $\cup$ tags
  end for
  return outputTagsOfAllLayers
end procedure

$$loss = \sum_{i=1}^{n} loss_i \tag{3.1}$$

The training loss of Multi-BiLSTM-CRF model is computed from the loss of all its layers (Eq. 3.1). The loss of each layer is calculated in the same way as the loss of a BiLSTM-CRF model presented in Section 3.3.1. Multi-BiLSTM-CRF is also trained as a normal neural network which uses back-propagation and gradients to update network parameters that minimize the value of the loss function.

3.3.4 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. 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 (section 3.3.2) are comparable. Consequently, the training time is not reduced significantly and the performance of these

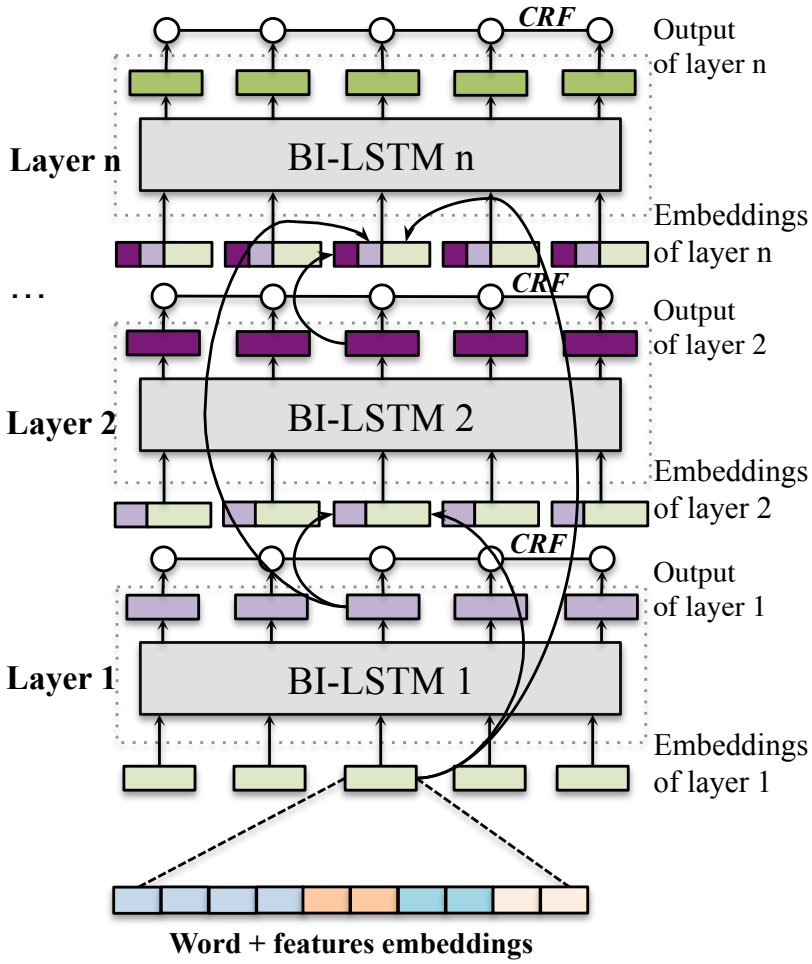


Figure 3.4: The multilayer BiLSTM-CRF model to recognize overlapping RE parts. Each BiLSTM-CRF component will predict tags at a layer and compute the tag score vectors for higher components. This model can be used for n-layer sequence labeling tasks.

two models are quite comparable. Secondly, in Multi-BiLSTM-CRF, the input sentence has been encoded many times in the same way by BiLSTM components. This cause some ineffectiveness 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 Figure 3.5.

A Multi-BiLSTM-MLP-CRF 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].

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

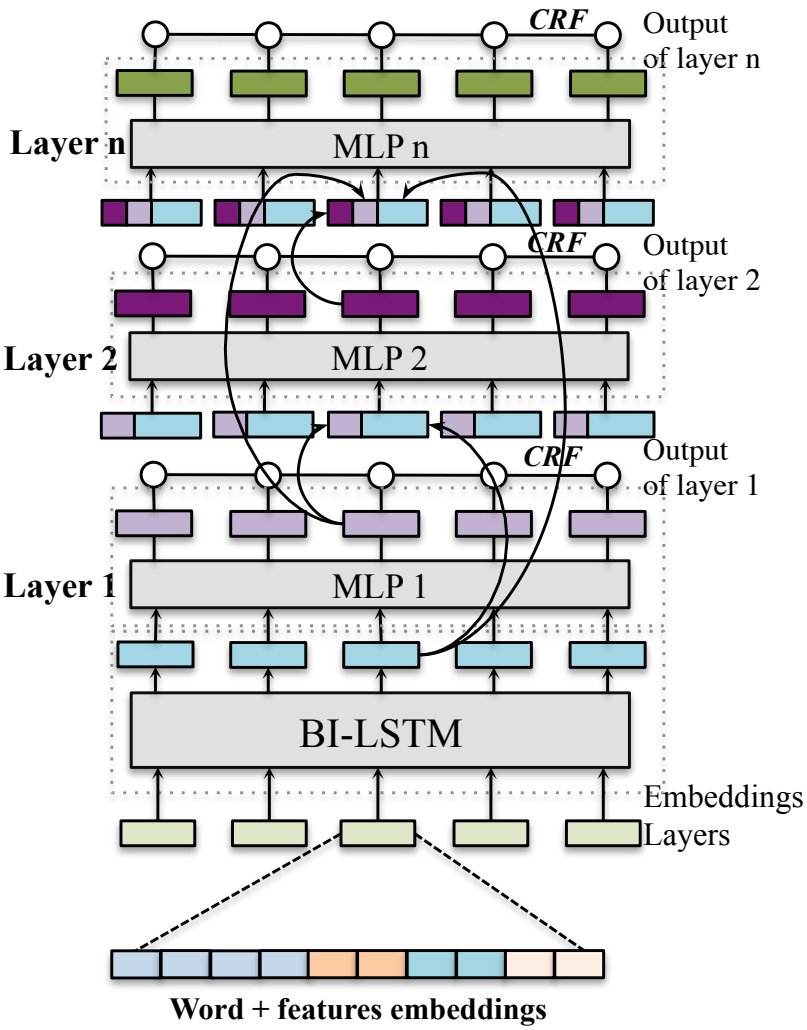


Figure 3.5: The multilayer BiLSTM-MLP-CRF model to recognize overlapping RE. The BiLSTM component encodes the input sequence that produces hidden vectors for MLP components to predict the labels. Each MLP component of each layer will compute tag scores for higher layers and predict tags at that layer.

3.4 Experiments

3.4.1 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 (Section 3.3.1).

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 3.3 and their statistics are shown in Table 3.5. 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 categorical values, and an example of these features is shown in Table 3.4. These features are expected to encourage the deep learning models to recognize the boundary of RE parts better.

Table 3.4: An example of the feature exaction step in JCC-RRE dataset. 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; or B-IF, I-IF, E-IF indicates word is the begin, inside and the end of a *If* clause

Features	**If**	**an**	**heir**	**dies**	**without**	**having**	**made**	**acceptance**	**..**
* POS	IN	DT	NN	VBZ	IN	VBG	VBN	NN	..
* Verb and noun phrases	-	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	-	-	-	-	-	-	-	-	..

3.4.2 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

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

Table 3.5: 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

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:

$$precision = \frac{\#\text{correct parts}}{\#\text{predicted parts}}, recall = \frac{\#\text{correct parts}}{\#\text{actual parts}} \tag{3.2}$$

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{3.3}$$

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.

3.4.3 Experimental Setting and Design

We conducted experiments on two 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] and 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 [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.

3.4.4 Results

Results on the JPL-RRE dataset Table 3.6 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 3.7 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 (section 3.3.2). Last three models (4,5,6) are multilayer models which are described in section 3.3.3 and 3.3.4. 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 Figure 3.6. In addition, the detailed results of each label are presented in Table 3.8.

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

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

Table 3.6: Experimental results on the Japanese National Pension Law RRE datasets with different feature sets. Results of CRF models (1-4) are from Ngo et al. [2010] and Nguyen et al. [2011]

	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%**

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 3.7, 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 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

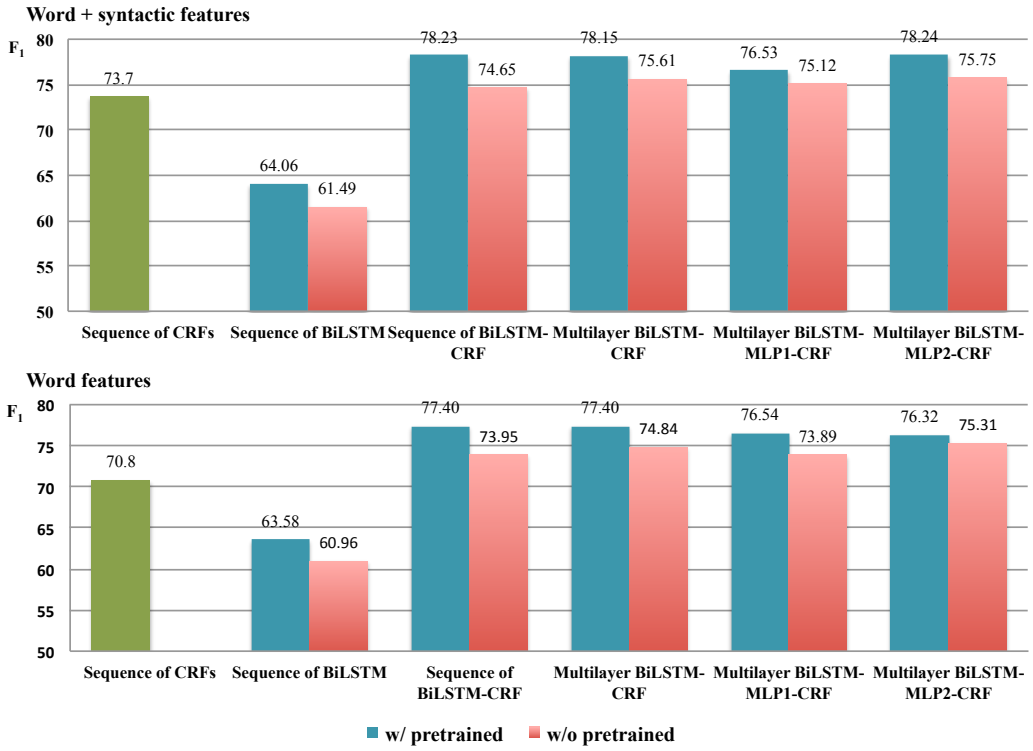
Table 3.7: Experimental results (F_1 score) on JCC-RRE dataset using end-to-end evaluation method. All models are trained and evaluated using 10-fold cross validation with the same training sets and test sets. The bold and italic numbers denote the best and the second best results.

Models	**Pre. emb**	**Features**	**Layer 1**	**Layer 2**	**Layer 3**	**All layers**
(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**

Table 3.8: 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

Figure 3.6: Comparison between different models on JCC-RRE dataset



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 3.9). 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.

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 3.9. 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.

Figure 3.7 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

Table 3.9: Number of parameters, training time (per epoch), testing time of all models in JCC-RRE data set. 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

Model	**#params**	**Training time / epoch**	**Testing time (1660 sentences)**	F_1 **score**
Multi-BiLSTM-CRF	650k	126 s	48.2 s	78.15
Multi-BiLSTM MLP1-CRF	213k	51 s	20.8 s	76.53
Multi-BiLSTM MLP2-CRF	240k	55 s	21.2 s	78.24
Sequence of BiLSTM-CRF	654k	140 s	52.9 s	78.23

Figure 3.7: 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

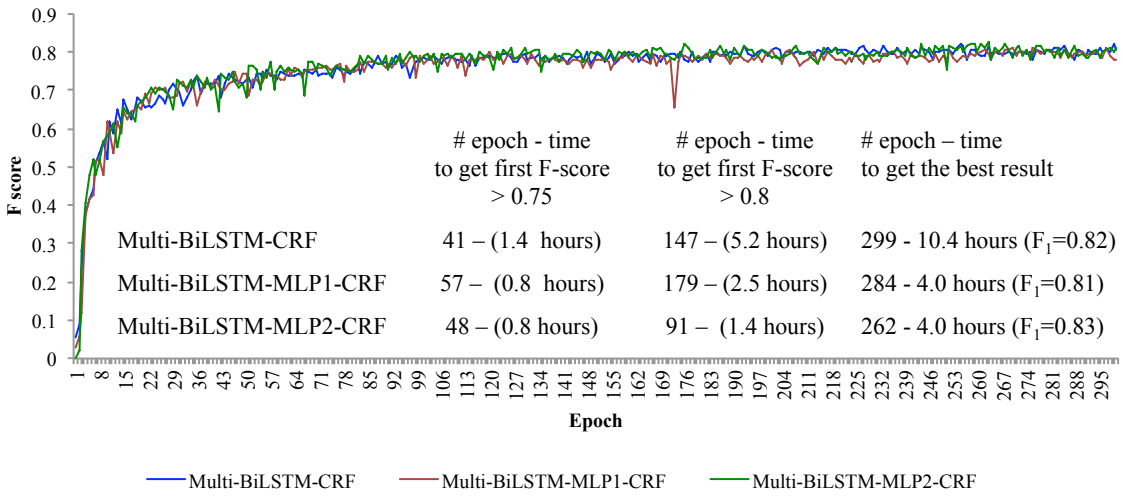


Table 3.10: Comparison between end-to-end evaluation and single-evaluation method on JCC-RRE dataset

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

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 3.7 . 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 3.10 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 contain only *Unless* parts and recognizing these parts do not depend on tags of other layers).

3.4.5 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 predict the boundaries of RE parts better. For example, table 3.11 shows outputs of the sequence of BiLSTM-CRF models of two cases: (a) w/o features and (b) w/features. In case (a), without

Table 3.11: 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
neighborin									
g	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					
,	,	-	-	-					

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 3.12 shows an example in 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.

Table 3.12: Output of our the sequence of BiLSTM-CRF models 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
.	.	-	-	-	-	-	-	-	-	-

The effects of sentence length on performance: Table 3.13 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 sentences. However, in short sentences, requisite parts usually appear in preposition phrases that may cause some difficulties for recognizing RE parts. Table 3.14 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 3.15 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

Table 3.13: 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**

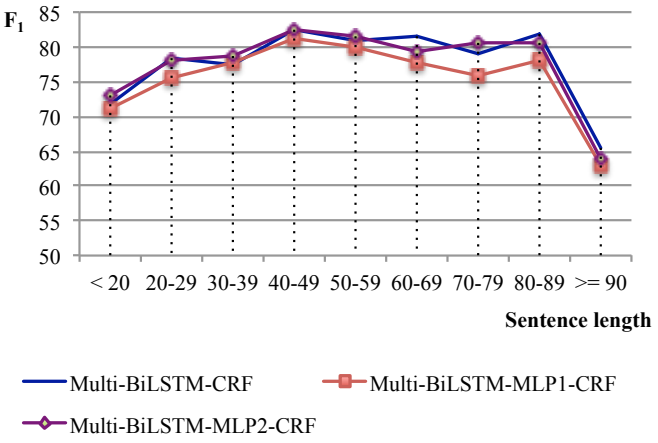


Table 3.14: Some sample outputs of our model (Multi-BiLSTM-CRF) on short sentences. In case (a), our model correctly predicts both R and E parts; in case (b), our model predicts the R part correctly, but not the E part; in cases (c) and (d), our model failed to predict both the R and E parts.

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	-
.	-	-	-	-

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 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 3.15: 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 3.8

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

3.5 Conclusions and Future work

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.

There are two directions for our future work related to the RRE task. 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, the proposed models can be applied to other sequence labeling tasks in different domains such as named entity recognition, information extraction, semantic role labeling or discourse parsing. 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.

Chapter 4

Recognizing Textual Entailment in Legal Texts

This chapter presents our study on recognizing textual entailment in legal texts. Firstly, in Section 4.1, we present the importance of the RTE in legal texts, previous approaches for this task and their limitations. We also present the motivation of our study and research objectives. The detailed description of the COLIEE entailment task and the dataset are described in Section 4.2. We then present deep learning models which are used for the COLIEE entailment task in Section 4.3. In Section 4.4, we present a semi-supervised approach to tackle the lack of labeled data problem with two methods for data augmentation using syntactic parse trees and requisite-effectuation structures. Section 4.5 presents the new approach which are based on sentence decomposition. The proposed model, called Multi-Sentence Entailment Model, also is described. We next describe experiments and results in Section 4.6. Finally, conclusions and future work are presented in Section 4.7.

4.1 Introduction

Recognizing entailment in legal texts is one of the important tasks in the Legal Engineering Field, an engineering approach to laws in e-Society Age [Katayama, 2007]. This task benefits systems such as Question Answering, and Summarization and other information systems.

A yes/no legal answering system is a good example to show the importance of this task. In legal domain, there are many statements/questions which we want to check their correctness. Given a statement, the question answering task must answer whether or not it is correct. For example, we want to check the correctness of the statement "*The family court may order the commencement of curatorship without the consent of the person in question*". It is easy to recognize that the entailment task is the core component for this QA system. If the given statement is entailed from legal articles, it is correct, otherwise incorrect. Building such Yes/No Question Answering system is an aim of the Competition on Legal Information Extraction/Entailment (COLIEE) [Kim et al., 2016b, Kano et al., 2017b]. In COLIEE, recognizing textual entailment task is one of the most important tasks besides the retrieval task which retrieves relevant articles for the input statement.

Recognizing textual entailment in legal texts also plays an important role in a legal knowledge management system which can help law experts to check whether or not a

newly enacted article is conflict or redundant with existing articles. If a statement in a new enacted articles is entailed from existing articles, this statement may be a redundancy; or, if that statement is contradicted (one type of "*not entailed*" relationship) from it related articles, this statement may be a conflict. In this case, the new article should not be enacted. Law experts or people in this domain benefit from these systems because checking the conflicts and redundancies manually is very time-consuming.

This task is a type of recognizing textual entailment task or Natural language inference task which has been described in Chapter 2. Therefore, it can be solved by using the techniques of RTE and NLI. This study mainly focuses on the entailment task in COLIEE and all experiments are conducted on COLIEE benchmark datasets from COLIEE 2014 to 2017. A clear description of the task is presented in Section 4.2. Previous works used both supervised and unsupervised learning methods for recognizing the entailment relationship between a question and its relevant articles.

- *Unsupervised learning approaches*: Kano et al. [2017a] employed a case-role analysis step to extract subject-predicate pairs in questions and articles. Subjects and predicates in a given question and its related articles are matched together to determine the entailment result using a threshold and the support of a list of heuristic rules. Although the method is unsupervised, it still needs the training data to tune the threshold. In addition, the usage of heuristic rules and case-role analysis tools shows that this method is language-dependent.

- *Supervised learning approaches*: Previous works use on both conventional machine learning models and deep learning models by treating this task as a classification problem. Do et al. [2016] employed a convolutional neural network (CNN) with the input features is the combination of word embedding, LSI, TF-IDF score. Kim et al. [2016a] utilized a variety of similarity features and three different classifiers (decision tree, linear SVM, CNN) to classify an input pair. A classification decision is obtained by using majority vote. Some modern techniques of deep learning are also adapted for the task such as the attention mechanism [Morimoto et al., 2017, Nguyen et al., 2017].

Previous approaches exhibit several limitations. Some methods (e.g. [Kano et al., 2017a]) still depend on handcraft features and heuristics, which are based on the analysis of the language and the training data, so it is difficult to apply to legal texts in other languages. Besides, the COLIEE entailment task is very challenging because semantics analysis is one of the most difficult tasks in NLP but the COLIEE dataset is small and sentences are very long and complex. Consequently, supervised learning algorithms suffer from insufficient training data and they cannot obtain enough knowledge to train good models. In some preliminary experiments in our study, although our models are trained using some state-of-the-art deep learning models in the natural language inference task, the trained models cannot capture some simple cases although human can easily recognize them such as negations or contradictions.

In this study, we first apply deep learning approaches for recognizing textual entailment in legal texts. We then propose a semi-supervised approach to deal with the lack of labeled data problem. Our method exploits structures of a legal sentence to construct weak labeled data from a legal corpus. Two type of structures in a legal sentence is used. They are syntactic structures which based on the parse trees and logical structures

which based on requisite and effectuation parts. While parse trees can be obtained using standard language parser such as Stanford Parser [Klein and Manning, 2003], requisite and effectuation parts are obtained by analyzing an annotated corpus or by employing a pre-trained RE parser from Chapter 3. The weak labeled dataset then is combined with the original dataset to train entailment classifiers.

In COLIEE entailment task, we observed that a complex sentence might contain several single statements; and the entailment decision can be identified by using a few of them. Therefore, we proposed a model to decompose a long sentence into simple statements/sentences. This step will decompose the relevant articles in the input pair into a list of simple statements/sentences. To predict the entailment between the question and its relevant articles, we propose a novel model for entailment classification, called Multi-Sentence Entailment Model (MSEM), that can handle relevant articles as a list of sentences instead of a very long sentences as previous approaches.

Experimental results on two official test sets H27 and H28 of the COLIEE 2016 and 2017 [Kim et al., 2016b, Kano et al., 2017b] show that our new augmented datasets yield positive effects. Firstly, the performance of trained models on the new dataset has significant improvements in comparison with models trained on the original data set. Secondly, models which are trained on the new datasets are more stable because they do not bias on the majority classes. Besides, it can predict some entailment phenomena such as negation or sub-sentences. The performance of MSEM with sentence decomposition is also comparable with best systems. Our systems outperform previous baselines in both of two benchmark test sets.

The details of the COLIEE entailment task, deep learning models for the entailment task in legal texts, the semi-supervised approach, the sentence decomposition and the Multi-Sentence model are described in next sections.

4.2 The COLIEE Entailment task

The entailment task is the second task in COLIEE beside the retrieval task. Both of these two tasks are combined to build an end-to-end question answering system that can answer Yes/No questions in Japanese Legal Bar exams.

Task definition: Given a question (or a statement) Q and a set of relevant articles $\{S_1, S_2, \ldots, S_n\}$. The entailment task will determine whether or not Q is entailed from $\{S_1, S_2, \ldots, S_n\}$. If the question Q is entailed from its relevant articles, the answer is YES, otherwise NO. In the context of the end-to-end question answering system, the set of relevant articles of the given question is retrieved by the retrieval task. However, in this task, relevant articles of a question have been provided. Table 4.1 shows an example of the COLIEE's entailment task. By considering the content of related articles as a **text**, and the content of the given question is **hypothesis** we can solve this task using approaches for NLI and RTE which had been described in Section 2.

The COLIEE entailment dataset: Questions in the COLIEE entailment corpus is drawn from Japanese national bar examinations [1], the related articles of each question

[1]National bar exam is an exam which attorneys at law are required to pass `https://en.wikipedia.org/wiki/Attorneys_in_Japan`

Table 4.1: An example of the COLIEE's entailment task. The input is a pair of question (or statement) and its relevant articles. An entailment system will check whether or not the given statement is inferred/entailed from its relevant articles

Articles	With respect to any person who whose capacity is extremely insufficient to appreciate right or wrong due to any mental disability, the family court may order the commencement of curatorship upon a request by the person in question, his/her spouse, any relative within the fourth degree of kinship, the guardian, the supervisor of the guardian, the assistant, the supervisor of the assistant, or a public prosecutor;provided however, that, this shall not apply to any person in respect of whom a cause set forth in Article 7 exists.
Statement	The family court may order the commencement of curatorship without the consent of the person in question.
Entailment	Yes

and the label of entailment relationship has been provided. Therefore, each instance in the training dataset is a triple of a question, a set of relevant articles and an entailment relationship label (YES or NO). Compared to NLI task, the COLIEE task is very challenging because the size of the dataset is very small. Besides, sentences in the COLIEE dataset is very long and complex; the entailment relationship is not only based on the analyzing of a single but a list of sentences. The comparison and some examples of two data sets are shown in Tables 4.2 and 4.3.

Table 4.2: Examples of existing RTE and NLI datasets

RTE [Dagan et al., 2006]

Text	The Republic of Yemen is an Arab, Islamic and independent sovereign state whose integrity is inviolable, and no part of which may be ceded.
Hypothesis	The national language of Yemen is Arabic.
Output	Yes

NLI [Bowman et al., 2015]

Text	A man inspects the uniform of a figure in some East Asian country.
Hypothesis	The man is sleeping
Output	Contradiction

Evaluation metrics The evaluation measure is the accuracy which measures the percentage of questions which are answered correctly. It is calculated using formula 4.1.

$$Accuracy = \frac{\#\text{queries which were answered correctly}}{\#\text{all queries}} \tag{4.1}$$

4.3 Recognizing textual entailment using sentence encoding-based and attention-based models

In this section, we denote the input of the entailment task is a pair of a question and its relevant articles. Let **a** is the text represented for related articles and **b** is the text

Table 4.3: Comparison between COLIEE dataset and SNLI dataset

	COLIEE entailment task (**COLIEE corpus**)	Natural language inference (**SNLI corpus**)
Corpus size	500 pairs	550, 000 pairs
Sentence length	Long (from 10 - n x 100 words)	Short (from 8-15 words)
Complexity	High: -Semantic analysis between a sentence and related articles which consists of multiple sentences. -Syntactic structures are complex	Low: -Semantic analysis between 2 simple sentences Syntactic structures are simple
	-Abstract / concrete	

represented for the question. **a** and **b** are two sequences of words which are considered as two sentences. A word in is represented as a vector, called word embedding vector. Below are notations used in this section:

- $\mathbf{a} = (a_1, a_2, \ldots, a_{l_a})$ is a sequence of vectors represented for the text in related articles.
- $\mathbf{b} = (b_1, b_2, \ldots, b_{l_b})$ is a sequence of vectors represented for the question.
- a_i, b_j are d-dimensional embedding vectors of word i in **a** and word j in **b**.
- l_a and l_b are the length of **a** and **b**, respectively.

4.3.1 Sentence Encoding-Based Models

Figure 4.1 shows the architecture of the sentence encoding-based model used in our experiments. The model consists of several components as follows:

Encoding: We use CBOW method or a BiLSTM to encode sequence of words of a sentence into a vector.

- CBOW: The vector representation of a sequence is a summation of all word embeddings in that sequence as the following equations:

$$\mathbf{v}_a := \sum_{i=1}^{l_a} a_i \qquad \mathbf{v}_b; = \sum_{j=1}^{l_b} b_j \tag{4.2}$$

- BiLSTM: In this method, a BiLSTM is used to encode a sequence into a list of hidden states. We then use the last hidden state as the vector representation of the sequence. In this case, $\mathbf{h}_{a,l_a}$ and $\mathbf{h}_{b,l_b}$ are last hidden states of **a** and **b**

$$\begin{aligned} \mathbf{h}_{a,i} &:= \text{BiLSTM}(\mathbf{a}, i) \quad \forall i \in [1, ..., l_a] \\ \mathbf{h}_{b,j} &:= \text{BiLSTM}(\mathbf{b}, j) \quad \forall j \in [1, ..., l_b] \end{aligned} \tag{4.3}$$

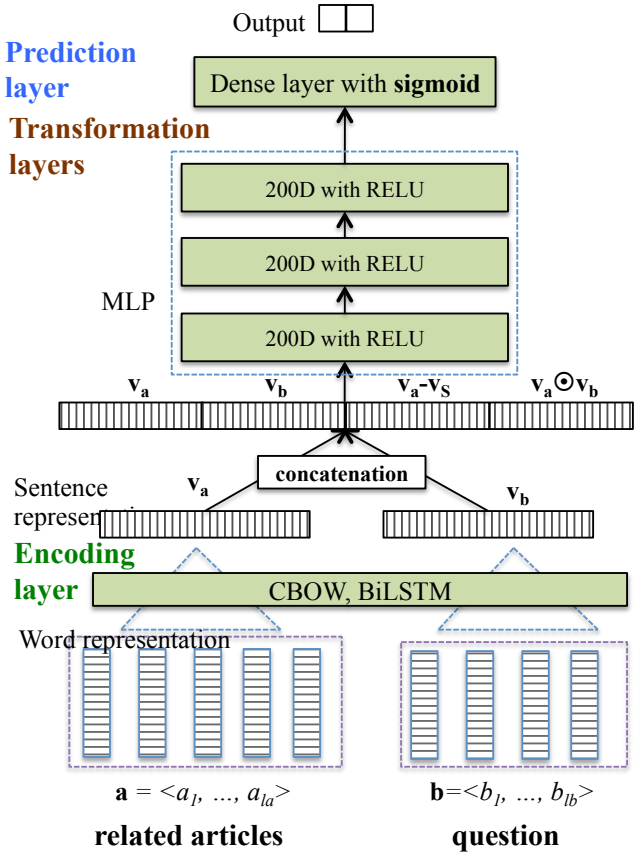


Figure 4.1: The sentence encoding model for recognizing the entailment between a question and the relevant articles

$$\begin{aligned} \mathbf{v}_a &:= \mathbf{h}_{a,l_a} \\ \mathbf{v}_b &:= \mathbf{h}_{b,l_b} \end{aligned} \tag{4.4}$$

After the encoding step, $\mathbf{v}_a$ and $\mathbf{v}_b$ are vector representations of $\mathbf{a}$ and $\mathbf{b}$.

Transformation and Prediction : After two texts are encoded, these two vectors combined into a single vector:

$$\mathbf{v} = [\mathbf{v}_a; \mathbf{v}_b; \mathbf{v}_a - \mathbf{v}_b; \mathbf{v}_a \odot \mathbf{v}_b] \tag{4.5}$$

where $[\bullet, \bullet]$ denotes the concatenation and $\odot$ denotes the element-wise product of two vectors.

The vector $\mathbf{v}$ then is fed through a three-layer MLP with RELU activation functions. We also apply the dropout technique [Srivastava et al., 2014] after each layer in the MLP. Finally, the logistic regression layer will compute the entailment probability of the input pair using sigmoid function.

4.3.2 Decomposable attention models

Figure 4.2 shows the architecture of the decomposable attention model for natural language inference proposed by Parikh et al. [2016] which is used to recognize the relationship between two input texts decomposing the problem into sub-problems. This model is composed from three main components: **attend**, **compare** and **aggregate**.

Attend This component will soft-align the elements of **a** and **b** using the neural attention technique. For each word a_i in **a**, this step will find a sub-phrase β_i in **b** that soft-aligned to a_i and vice versa for α_j. The values of β_i and α_j are computed by equation 4.6 and 4.7.

$$\beta_i := \sum_{j=1}^{l_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_b} \exp(e_{ik})} b_j \tag{4.6}$$

$$\alpha_j := \sum_{i=1}^{l_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_a} \exp(e_{ik})} a_i \tag{4.7}$$

where e_{ij} are attention weights between words in **a** and **b** which is computed using a feed forward neural network F:

$$e_{ij} = F(a_i)^T F(b_j) \tag{4.8}$$

The obtained aligned phrases $\{(a_i, \beta_i)\}_{i=1}^{l_a}$ and $\{(b_j, \alpha_j)\}_{j=1}^{l_b}$ allow the model to decompose the problem into the comparison of aligned sub-phrases.

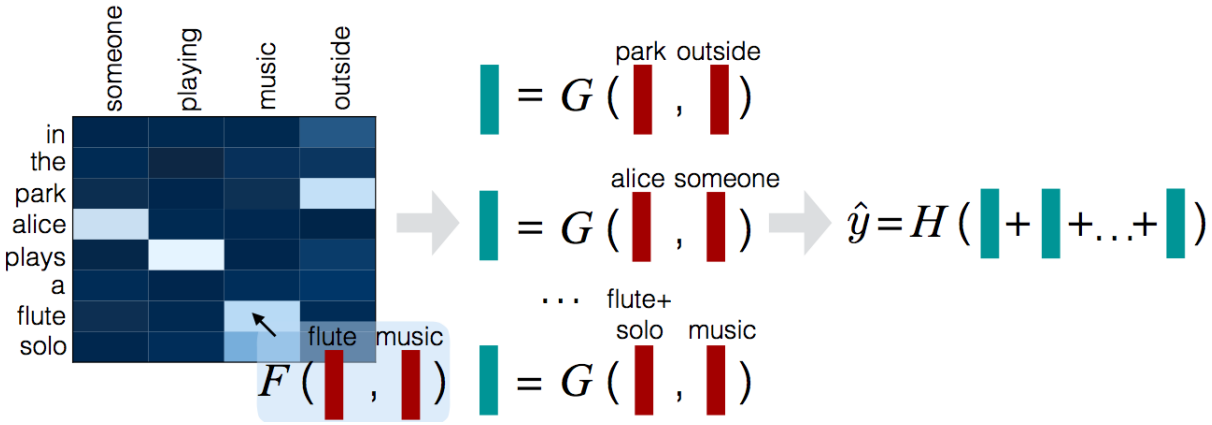


Figure 4.2: The decomposable attention model for recognizing textual entailment between two sentences including 3 steps: *Attend* (left), *Compare* (center) and *Aggregate* (right) [Parikh et al., 2016]

Compare This step will compare aligned phrases obtained from the previous step using a function G.

$$\begin{aligned} \mathbf{v}_{1,i} &:= G([a_i, \beta_i]) \quad \forall i \in [1, ..., l_a] \\ \mathbf{v}_{2,j} &:= G([b_j, \alpha_j]) \quad \forall j \in [1, ..., l_b] \end{aligned} \tag{4.9}$$

where the brackets $[\bullet, \bullet]$ denote the concatenation between two vectors. G is again a feed forward neural network with RELU activations.

Aggregate Two sets of comparison vectors $\{\mathbf{v}_{1,i}\}_{i=1}^{l_a}$ and $\{\mathbf{v}_{2,j}\}_{j=1}^{l_b}$ are aggregated into two vector $\mathbf{v}_1$ and $\mathbf{v}_2$ using summation:

$$\mathbf{v}_1 = \sum_{i=1}^{l_a} \mathbf{v}_{1,i} \qquad \mathbf{v}_2 = \sum_{j=1}^{l_b} \mathbf{v}_{1,i} \tag{4.10}$$

Finally, a classifier H is used to predict the scores for each class. H is a feed forward network followed by a linear layer:

$$\hat{y} = H([\mathbf{v}_1, \mathbf{v}_2]) \tag{4.11}$$

where $\hat{y} \in \mathbb{R}^C$ is a vector that represents the scores of each classes (e.g. Yes/No). Then, the predicted class is computed by $\mathbf{argmax}_i \hat{y}_i$.

For training, the model is trained using dropout regularization with the cross-entropy loss function. During the training process, parameters of F, G, H is updated to minimize the loss function.

4.3.3 Enhanced Sequential Inference Model

Enhanced Sequential Inference Model (ESIM) is proposed by Chen et al. [2016]. This model is also an attention-based model, but it has several differences from the decomposable attention model. First, two sequences of hidden states of text and hypothesis are obtained using BiLSTM in which each hidden state represents a word and its context. Hidden states of two sequences then are compared and aligned. The alignments information then will be passed into higher layers to aggregate and make the entailment decision. In ESIM, the alignment between words in two texts is computed based on hidden states instead of computing directly based on word embeddings as the decomposable attention model. Below are the details of the model.

Input: Given the text $\mathbf{a} = (a_1, a_2, ..., a_{l_a})$ and hypothesis $\mathbf{b} = (b_1, b_2, ..., b_{l_b})$, ESIM consists of several components as follows:

Input encoding: Both of $\mathbf{a}$ and $\mathbf{b}$ are encoded into lists of hidden states using a BiLSTM. $\bar{a}_i$ and $\bar{b}_j$ are hidden states represented for word i in $\mathbf{a}$ and word j in $\mathbf{b}$, respectively.

$$\begin{aligned} \bar{a}_i &:= \text{BiLSTM}(\mathbf{a}, i) \quad \forall i \in [1, ..., l_a] \\ \bar{b}_j &:= \text{BiLSTM}(\mathbf{b}, j) \quad \forall j \in [1, ..., l_b] \end{aligned} \tag{4.12}$$

Local inference modeling (or attention): For each word in a_i, this step will find words in $\mathbf{b}$ which are aligned with a_i. These words is represented by represented by $\tilde{\mathbf{a}}_i$. In equation 4.14, the normalization value $\frac{\exp(e_{ij})}{\sum_{k=1}^{l_b} \exp(e_{ik})}$ is represented for the alignment score between a_i and b_j. Finding words in $\mathbf{a}$ which are aligned with each word in $\mathbf{b}$ is the same way (see Equation 4.15).

$$e_{ij} = \bar{\mathbf{a}}_i^\top \bar{\mathbf{b}}_j \tag{4.13}$$

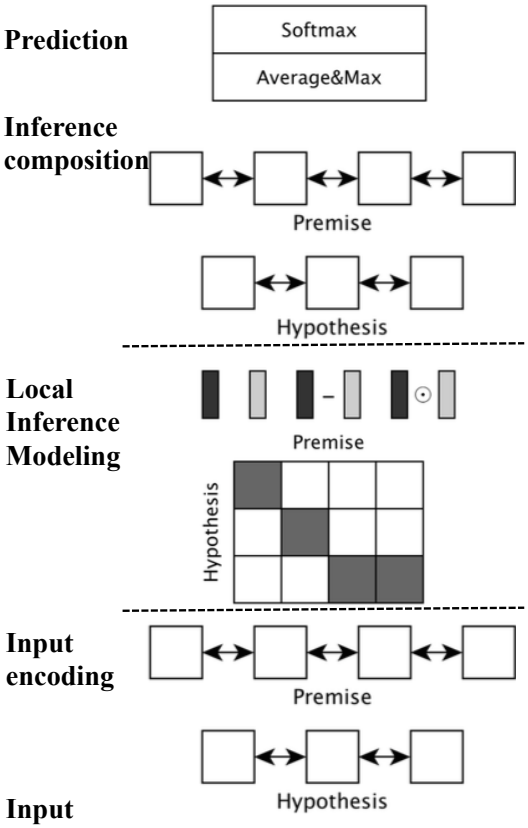


Figure 4.3: The Enhanced Sequential Inference Model (ESIM)

$$\tilde{\mathbf{a}}_i := \sum_{j=1}^{l_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_b} \exp(e_{ik})} \bar{\mathbf{b}}_j \tag{4.14}$$

$$\tilde{\mathbf{b}}_j := \sum_{i=1}^{l_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_a} \exp(e_{ik})} \bar{\mathbf{a}}_i \tag{4.15}$$

All word representations, local inference information are collected into $\mathbf{m}_a$ and $\mathbf{m}_b$ which is are sequences of vectors represented for text and hypothesis.

$$\mathbf{m}_a = \{\mathbf{m}_{a,i}\}_{i=1}^{l_a} \qquad \mathbf{m}_{a,i} = [\bar{\mathbf{a}}_i; \tilde{\mathbf{a}}_i; \bar{\mathbf{a}} - \tilde{\mathbf{a}}_i; \bar{\mathbf{a}}_i \odot \tilde{\mathbf{a}}_i] \tag{4.16}$$

$$\mathbf{m}_b = \{\mathbf{m}_{b,j}\}_{j=1}^{l_b} \qquad \mathbf{m}_{b,j} = [\bar{\mathbf{b}}_j; \tilde{\mathbf{b}}_j; \bar{\mathbf{b}}_j - \tilde{\mathbf{b}}_j; \bar{\mathbf{b}}_j \odot \tilde{\mathbf{b}}_j] \tag{4.17}$$

Inference composition: The composition layer also employed BiLSTMs to transform $\mathbf{m}_a$ and $\mathbf{m}_b$ into lists of hidden states (Equation 4.18). The obtained vectors $\mathbf{v}_a$ and $\mathbf{v}_b$ represent for the composition information. They are then aggregated into a single vector $\mathbf{v}$ using average pooling, max pooling, and concatenation followed equations 4.19 and 4.20.

$$\begin{aligned} \mathbf{v}_{a,i} &:= \text{BiLSTM}(\mathbf{m}_a, i) \qquad \forall i \in [1, ..., l_a] \\ \mathbf{v}_{b,j} &:= \text{BiLSTM}(\mathbf{m}_b, j) \qquad \forall j \in [1, ..., l_b] \end{aligned} \tag{4.18}$$

$$\begin{aligned} \mathbf{v}_{a,\text{ave}} &:= \sum_{i=1}^{l_a} \frac{\mathbf{v}_{a,i}}{l_a} \qquad \mathbf{v}_{a,\text{max}} := \max_{i}^{l_a} \mathbf{v}_{a,i} \\ \mathbf{v}_{b,\text{ave}} &:= \sum_{j=1}^{l_b} \frac{\mathbf{v}_{b,j}}{l_b} \qquad \mathbf{v}_{b,\text{max}} := \max_{j}^{l_b} \mathbf{v}_{b,j} \end{aligned} \tag{4.19}$$

$$\mathbf{v} = [\mathbf{v}_{a,\text{ave}}; \mathbf{v}_{a,\text{max}}; \mathbf{v}_{b,\text{ave}}; \mathbf{v}_{b,\text{max}}] \tag{4.20}$$

Prediction: The vector $\mathbf{v}$ then is put into a multilayer perceptron classifier. The MLP has a hidden layer with *tanh* activation. Finally, a $softmax$ output layer is used to produce the final prediction.

4.4 A Semi-supervised Approach for RTE in Legal Texts

In this section, we propose a simple semi-supervised learning approach to deal with the lack of labeled data problem. This approach consists two steps. After using an unsupervised method for data augmentation, the supervised learning methods with deep learning models are used to train entailment classifiers.

4.4.1 Unsupervised methods for data augmentation

Based on syntactic parse trees From the syntactic trees of sentences in Japanese Civil Code, we extract sub-sentences and negations of sub-sentences to add new instances to enrich the training corpus. The input, output and the rule for identifying the labels of generated instances are as followed:

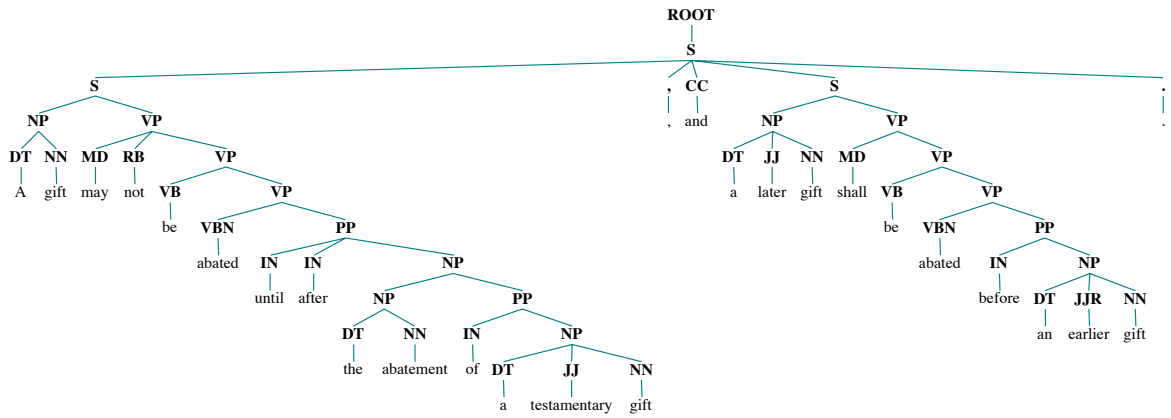


Figure 4.4: The parse tree of the sentence "A gift may not be abated until after the abatement of a testamentary gift, and a later gift shall be abated before an earlier gift."

- Input: A sentence $\mathbf{t}$ in Japanese Civil Code

Table 4.4: Four new training instances $\{(\mathbf{t}, h_1, y_1), \therefore, (\mathbf{t}, h_4, y_4)\}$ are generated by analyzing the syntactic parse tree of figure 4.4 ."

Input	**t** = A gift may **not** be abated until after the abatement of a testamentary gift, and a later gift shall be abated before an earlier gift .	
Output	$\mathbf{h_1}$ = A gift may **not** be abated until after the abatement of a testamentary gift (sub sentence)	$\mathbf{y_1}$ = YES
	$\mathbf{h_2}$ = a later gift shall be abated before an earlier gift (sub sentence)	$\mathbf{y_2}$ = YES
	$\mathbf{h_3}$ = A gift may be abated until after the abatement of a testamentary gift (negation)	$\mathbf{y_3}$ = NO
	$\mathbf{h_4}$ = a later gift shall **not** be abated before an earlier gift (negation)	$\mathbf{y_4}$ = NO

- Output: A set of instances $\{(\mathbf{t}, h_1, y_1), (\mathbf{t}, h_2, y_2), ...\}$. Where each instance is a triple $(\mathbf{t}, h_i, y_i)$:
  - $\mathbf{t}$: the original sentence
  - h_i : a generated sentence
  - y_i : a YES/NO label that indicates the entailment relationship between $\mathbf{t}$ and h_i. If h_i is a sub-sentence of $\mathbf{t}$, the value of y_i is **YES**. Otherwise, if $\mathbf{h}$ is a negation of a sub-sentence of $\mathbf{t}$, the label y_i is **NO**

Generating sub-sentences of a sentence is based on the analysis on its syntactic parse tree with some following simple rules:

- Rule for generating a sub-sentence: Firstly, if the original sentence $\mathbf{t}$ is a simple sentence, the sub-sentence is the same to the original sentence. We use some patterns such as "NP VP ." and "PP , NP VP ." to extract simple sentences. Secondly, if the original sentence is a compound sentence that is composed of several independent sentences, we extract each independent sentence as a sub-sentence. We use patterns such as "S CC S .", "S , CC S ." and "S : S ." to extract compound sentences.
- Rule for generating a negation: After a sub-sentence is identified, we analyze the parse tree of sub-sentence and find the modal verb and the main verb in the main clause, we then add "not" after the modal verb to create the negation.

For example, figure 4.4 shows a parse tree of the compound sentence *"A gift may not be abated until after the abatement of a testamentary gift, and a later gift shall be abated before an earlier gift ."*. From this parse tree, we can identify two sub-sentences: *A gift may not be abated until after the abatement of a testamentary gift* and *a later gift shall be abated before an earlier gift*. Then, we generate two negations from these sub-sentences and make four new training instances to add to our current corpus. Table 4.4 shows four new instances which are generated from the input sentence.

Based on requisite-effectuation structures : to increase the diversity of generated sentences, we utilize the requisite-effectuation (R-E) structure of legal sentences to create

new training instances. An R-E structure consists of requisite parts and an effectuation part in which the requisite parts describe conditions and the effectuation part describes the effect of those conditions. Intuitively, if we extract an R-E structure from a legal sentence, the corresponding statement of the R-E structure is entailed from the sentence. R-E parts can be identified by using a pre-trained RE parser in Chapter 3 or from an annotated corpus. The data augmentation process is described as follows:

- Step 1: From a legal sentence $\mathbf{t}$, we extract a set of requisite-effectuation pairs $\{(r_1, e_1), (r_2, e_2), \therefore, (r_n, e_n)\}$ from that sentence. Each pair (r_i, e_i) is a R-E structure. We then will generate positive hypotheses for each R-E structure (r_i, e_i) by using following templates:

$$h_1\text{: } \textbf{if } [r_i]\textbf{, } [e_i] \qquad \text{or} \qquad h_2\text{: } [e_i]\textbf{, if } [r_i]$$

- Step 2: We negate the effectuation parts by using the negation technique in previous section and create negative hypotheses by using following :

$$h_3\text{: } \textbf{if } [r_i]\textbf{, not } [e_i] \qquad \text{or} \qquad h_4\text{: } \textbf{not } [e_i]\textbf{, if } [r_i]$$

- Step 3: From the set of negative and positive generated hypotheses, we create new training instances $(\mathbf{t}, h_i, \text{"YES"})$ and $(\mathbf{t}, h_i, \text{"NO"})$ where $\mathbf{t}$ is the original sentence, h_i and h_j are a positive and a negative hypothesis in the set.

- Step 4: We also create a negative example such as $(\text{text} = h_3, \text{hypothesis} = h_1, \text{"NO"})$ or $(\text{text} = h_4, \text{hypothesis} = h_2, \text{"NO"})$.

To construct high-quality instances, the requisite part r in an (r, e) pair must be the condition of the effectuation part e. Otherwise, if the requisite part and the effectuation part of a pair a not related, the quality of generated instances is very low. To ensure the extracted (r,e) pairs are correct structures, we just only focus on sentences that have only one Requisite part or one Effectuation part. An example of this method is shown in table 4.5. After the augmented dataset is constructed, it is combined with the original dataset and deep learning models to train entailment classification models.

4.4.2 Sentence filtering

In the COLIEE dataset, each instance is a pair of a question and its relevant articles. Although the content of the articles is relevant to the question, many sentences in the articles are not related to the question. This causes adverse effects on the performance if we use the whole content of relevant articles to predict the entailment relationship for the given question.

The sentence filtering step will retain sentences which are most similar to the given question. Firstly, we extract sentences from the relevant articles; and convert them into vectors by using vector space model with TF-IDF weighting method. We then convert the given question into a vector by the same way. Then we compute the similarity of sentences in the article and the given question. Top k sentences which are most similar to the given question will be retained.

Table 4.5: Four new training instances $\{(\mathbf{t}, h_1, y_1), \therefore, (\mathbf{t}, h_4, y_4)\}$ are generated by analyzing requisite-effectuation structure of the input sentence"

Input	**t** = The owner of the land on the other side may use the dam under the preceding paragraph if he/she owns part of the land containing the stream .	
R-E pair sets	• r_1 : if he/she owns part of the land containing the stream e_1 : The owner of the land on the other side may use the dam under the preceding paragraph	
Output	**h_1** = if he/she owns part of the land containing the stream, the owner of the land on the other side may use the dam under the preceding paragraph	**y_1** = YES
	h_2 = the owner of the land on the other side may use the dam under the preceding paragraph, if he/she owns part of the land containing the stream	**y_2** = YES
	h_3 = if he/she owns part of the land containing the stream, the owner of the land on the other side may **not** use the dam under the preceding paragraph	**y_3** = NO
	h_4 = the owner of the land on the other side may **not** use the dam under the preceding paragraph, if he/she owns part of the land containing the stream	**y_4** = NO

4.5 Recognizing Textual Entailment Using Sentence Decomposition and Multi-Sentence Entailment Classification Model

Observations in COLIEE dataset: Each input is a pair of a question and related articles. Each article has many sentences which are represented in several paragraphs. Each article can be separated into a set of single statements. Besides, many entailment decisions can be made using only one or two statements or a part of related articles. Each statement can be represented in the following form:

⟨ A statement = CONDITION* EFFECTUATION ⟩

Limitation of previous works: Previous works consider the related articles as "a single very-long sentence" which is the concatenation of all sentences in related articles. However, this approach is not effective because long sentences cause negative effects to neural networks. The length of a sentence may exceed 1000 of words. The previous solution (in Section 4.4.2) removes unrelated sentences to make long sentences shorter by using the concatenation of k most similar sentences using cosine similarity. However, the limitation of this method is that the similarity of two sentences is computed based on a lexical matching of term vectors. Therefore, several sentences may be removed because they do not share common terms with the question but they are relevant to the question.

Proposed approach: We propose an approach including two steps to tackle limitations of previous approaches. First, we propose a method to decompose a long sentence into a

list of simple sentences or statements to make the RTE task easier. Secondly, we propose a method to handle the inference between a list of sentences in articles and a question. The difference between the proposed approach and previous approaches is illustrated in Figure 4.5.

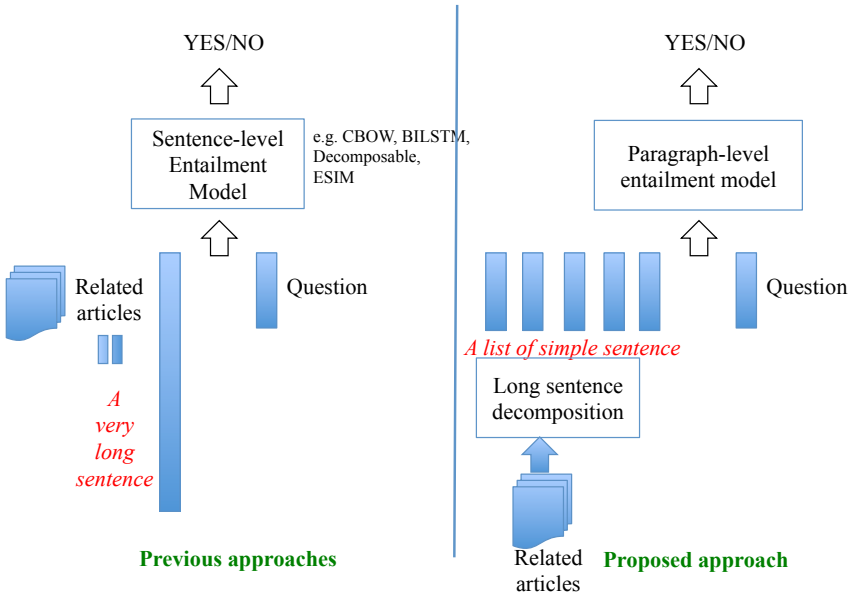


Figure 4.5: Comparison between previous approaches and the proposed approach

4.5.1 Article Decomposition

Purpose: The article decomposition step splits related articles into a list of simple sentences by decomposing every sentence in these related articles. The input and output of the long sentence decomposition task are as follows:

- *Input*: A long sentence
- *Output*: a list of simple sentences (each simple sentence is constructed by itemization detection, requisite-effectuation recognition)

Method 1: Using itemization detection We first detect itemized symbols and split the long sentence into a main sentence and a list of items by using regular expressions (e.g. "\s(?=[\:\;]\s\(\s?[ivx]+\s?\)"). We then detect a reference expression (e.g *the following persons*, *the following cases*, *the following items*) in the main sentence. Finally, we construct a list of simple sentences from the list, reference expression and the main sentence. Figure 4.6 shows the our process for long sentence decomposition by using itemization.

Method 2: Using requisite-effectuation structures To decompose a long sentence using RE structures, we first recognize RE parts in the sentences and identify a list of RE structures. In an RE structure, the effectuation part must be the effect of the requisite part. The list of simple sentences are then constructed from the list of RE structures using the following form:

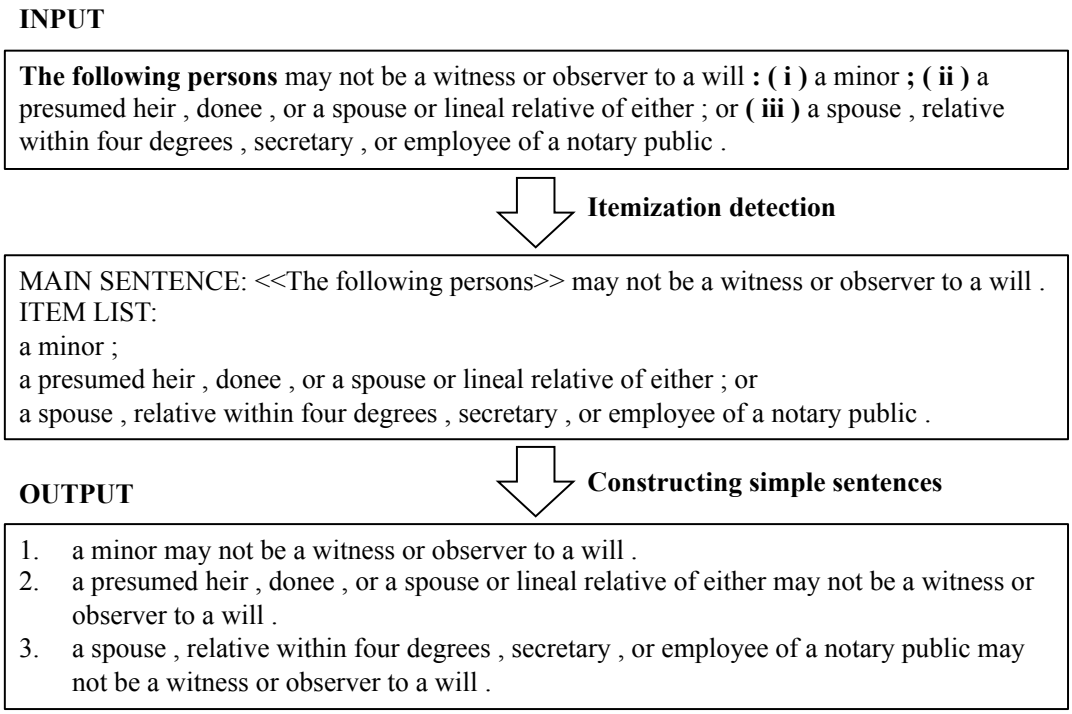


Figure 4.6: Long sentence decomposition using itemization detection

Input: a RE structure ($\mathbf{r}$, $\mathbf{e}$)
Output is a constructed sentence in the form of: if $\mathbf{r}$, $\mathbf{e}$

4.5.2 Multi-Sentence Entailment Classification Model

After sentences in related articles have been decomposed into a list of simple sentences, we propose a model to tackle with the inference between a list of sentences and a question which is considered as a sentence. Figure 4.7 shows the architecture of our model which includes important components such as Encoding layer, Sentence attention layer, Transformation layer and Prediction layer.

It assumes that the input is a pair of a question q and a list of simple sentences $\{s_1, s_2, \ldots, s_n\}$ in relevant articles. Firstly, sentences $\{s_1, s_2, \ldots, s_n\}$ are encoded into vectors $\{m_1, m_2, \ldots, m_n\}$ and the question q is encoded into vector u using the encoding layer which employs a method to encode a sequence of words such as CBOW or LSTM.

Secondly, in the sentence attention layer, we compute the match between u and each m_i by taking the inner product followed by a softmax. The meaning of this step is to get the attention vector p in which a value p_i is represented for the importance of each sentence s_i.

$$p_i = \text{softmax}(u^\top m_i) \tag{4.21}$$

where $\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$.

We then compute the vector representation of sentences o using the weighted sum between all sentence vectors and p as follows:

$$o = \sum_i p_i \mathrm{m}_i \tag{4.22}$$

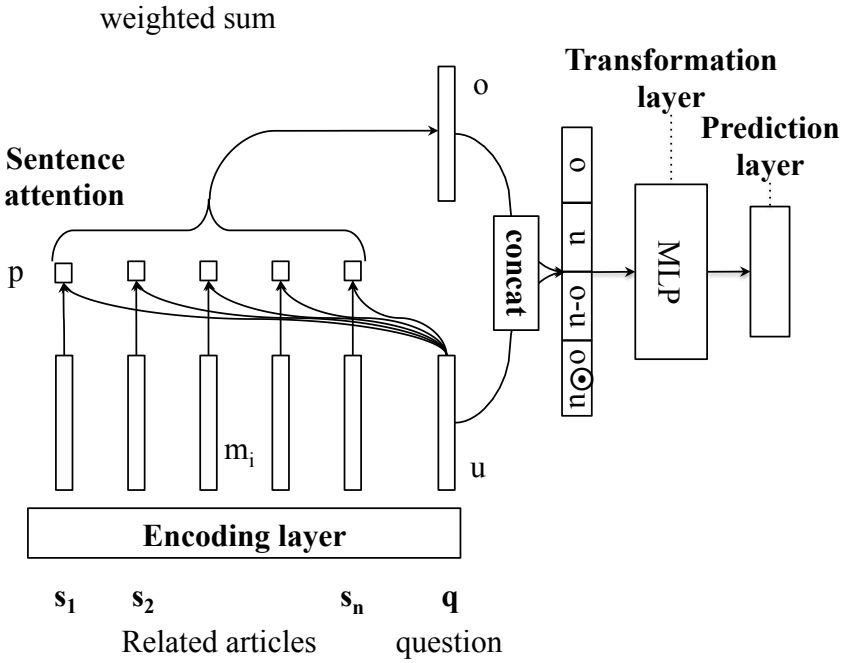


Figure 4.7: Paragraph-level entailment model based on article decomposition

The vector u and o then are concatenated to form the aggregation vector v. We also using difference and multiplication between u and o:

$$v = [o, u, o - u, o \odot u] \tag{4.23}$$

The vector **v** then is transformed by a transformation layer, which is an MLP with RELU activation functions. The MLP has one fully connected layer. The batch normalization technique [Ioffe and Szegedy, 2015] is also applied to normalize the output of each layer in the MLP. Finally, a fully connected layer with a sigmoid function at the prediction layer is employed to compute the entailment probability.

4.6 Experiments and Results

4.6.1 New Training Datasets

Data augmentation results From Japanese Civil Code which includes more than 1600 sentences, we use our proposed method (in Section 4.4) to generate three new datasets as below:

- *syntactic*: 2716 new training examples including negation and sub-sentence types are generated based on syntactic parse trees method.
- *recorpus*: 2160 new training examples are generated by analyzing RE structures in the JCC-RRE corpus.
- *reparser*: 2497 new training examples are generated by utilizing an RRE parser from Chapter 3. We employ a pre-trained model which is trained with Multilayer-BiLSTM-MLP-CRF to recognize RE parts.

Table 4.6: The statistic information of new training datasets. Dataset 1: the original dataset; Dataset 2 to 6: augmented datasets from the combination of the original dataset and generated datasets

Dataset	Number of instances	Description
Training		
* Dataset 1	507	*original*
* Dataset 2	5383	*original*+*syntactic*+*recorpus*
* Dataset 3	5720	*original*+*syntactic*+*reparser*
* Dataset 4	3223	*original*+*syntactic*
* Dataset 5	2667	*original*+*recorpus*
* Dataset 6	3004	original+*reparser*
Test		
* H27	74	official test set
* H28	78	official test set
* H27+278	152	combination of H27 and H28

Constructing new training datasets: After new training instances were generated, we combine them with the original dataset (*original*), which contains 507 instances, to create new datasets to train entailment classification models. The details of datasets used in our experiments are as follow:

- **Dataset 1** is the original COLIEE 2017 dataset [Kano et al., 2017b]. We use sets from H18 to H26 (507 instances) as the training set and the development set. Instances of H27 and H28 are used as the test set to evaluate the performance of classifier models. The development set is sampled randomly from the training set with the percentage of 20%. We also applied the sentence filtering process with **k**=2.

- **Dataset 2 to 6** are different combinations of the original dataset and different generated sets. From a generated set, we split them into two parts and add to the original training and development set to create a new dataset.

Note that three above datasets have the same test set which includes 152 instances (74 instances from H27 and 78 instances from H28). That allows us to evaluate the contribution of the newly generated dataset. The statistics of all training and test datasets are summarized in Table 4.6.

4.6.2 Experimental Results of Sentence Encoding-based Models and Attention-based Models

Model settings We modify some available tools [2] [3] to train sentence encoding-based and attention-based models. During the training process, although there are many hyper-parameters which we can tune to find the best configuration, we do not have the time to

[2] `https://github.com/erickrf/multiffn-nli`

[3] `https://github.com/NYU-MLL/multiNLI`

Table 4.7: Experimental results on the two test sets (H27 and H28) of models trained on Datasets 1 to 3.

			Dataset 1		**Dataset 2**		**Dataset 3**	
Model	Test	#ques.	acc.	avg. F1	acc.	avg. F1	acc.	avg. F1
Bi-LSTM	H27	74	0.4730	0.4268	**0.6757**[1]	**0.6754**[1]	**0.6351**[1]	**0.6335**[1]
CBOW			0.5270	0.5269	0.4595	0.4494	*0.6216*[2]	*0.6213*[2]
Decomposable			0.4730	0.4166	*0.6081*[2]	*0.6063*[2]	0.5405	0.5235
ESIM			0.4865	0.3273	0.5541	0.5533	0.5811	0.5804
Bi-LSTM	H28	78	0.4872	0.4817	**0.6410**[1]	*0.6253*[2]	*0.6026*[2]	*0.6025*[2]
CBOW			0.5385	0.5336	0.5256	0.4886	**0.6282**[1]	**0.6139**[1]
Decomposable			0.4615	0.4583	0.5897	0.5854	0.5897	0.5854
ESIM			0.6154	0.3810	**0.6410**[1]	**0.6401**[1]	0.5769	0.5752
Bi-LSTM	H27+H28	152	0.4803	0.4606	**0.6579**[1]	**0.6530**[1]	*0.6184*[2]	*0.6182*[2]
CBOW			0.5329	0.5319	0.4934	0.4714	**0.6250**[1]	**0.6202**[1]
Decomposable			0.4671	0.4470	*0.5987*[2]	0.5985	0.5658	0.5651
ESIM			0.5526	0.3559	*0.5987*[2]	0.5987[2]	0.5789	0.5789
Average result:			0.5082	0.4440	0.5872	0.5780	0.5970	0.5920

Table 4.8: Experimental results ($AvgF_1$) on the the combined test set (H27+H28) of different dataset combinations. Dataset 1 is the original dataset; Datasets 2 to 6 are augmented datasets. Each experiment is run in 5 times with different randomize initialization of parameters

Model	**Dataset 1**	**Dataset 2**	**Dataset 3**
* BiLSTM	0.3693 ± 0.03	**0.6080 ± 0.01**	**0.5863 ± 0.02**
* CBOW	0.5097 ± 0.02	0.5546 ± 0.01	0.5505 ± 0.01
* Decomposable Att.	0.3813 ± 0.04	0.5624 ± 0.02	0.5586 ± 0.02
* ESIM	0.4454 ± 0.06	0.5624 ± 0.02	0.5299 ± 0.01
Average:	0.4264 ± 0.04	0.5719 ± 0.02	0.5563 ± 0.02

Model	**Dataset 4**	**Dataset 5**	**Dataset 6**
* BiLSTM	*0.5787 ± 0.01*	**0.5654 ± 0.02**	**0.5645 ± 0.01**
* CBOW	0.5461 ± 0.03	0.5426 ± 0.01	0.5643 ± 0.03
* Decomposable Att.	**0.5900 ± 0.02**	0.5614 ± 0.03	0.5465 ± 0.02
* ESIM	0.5266 ± 0.02	0.4942 ± 0.01	0.5057 ± 0.02
Average:	0.5604 ± 0.02	0.5407 ± 0.02	0.5453 ± 0.02

tune all of them. We choose the word embedding size = 100, hidden layer size = 100, size = 8, drop-out rate = 0.2. All models are trained with Adagrad optimizer [Duchi

et al., 2011] and learning rate = 0.05. Each model is trained within 100 epochs, and we choose the one that produces the best performance on the development set. Besides, word embedding vectors are initialized from the pre-trained embedding source [Nguyen et al., 2017]. This embedding source is trained from a legal corpus by using word2vec tool [Mikolov et al., 2013].

Entailment classification results and discussions We train classifiers using four deep learning models on different datasets and evaluate trained models on benchmark test sets. Experiments are designed to evaluate the contributions semi-supervised approach. We also compare with participating systems in COLIEE 2016 and 2017. We use the *Accuracy* measure, average of F_1 ($avgF_1$) scores of two classes (YES and NO) to evaluate the performance.

Table 4.7 shows experimental results on first three datasets. The results demonstrate the contribution of the newly augmented datasets. Among three datasets, models which are trained on augmented datasets (Dataset 2 and 3) show the significant improvements. Overall, the average performance on models trained on Dataset 2 and 3 improves by nearly 10% in accuracy and 13% in F1 score compared to models trained on the original dataset (Dataset 1).

Table 4.9: Comparison with results of best systems reported in COLIEE 2016 (iLis7 [Kim et al., 2016a], KIS [Taniguchi and Kano, 2017], UofA [Kim et al., 2016c], N01 [Adebayo et al., 2016]) and COLIEE 2017 (KIS: [Kano et al., 2017a], NAIST [Morimoto et al., 2017], UA [Kim and Goebel], iLis). Symbols * indicate the previous state-of-the-art systems in English test sets

H27				**H28**		
System	**Accuracy**	**Langugage**		**System**	**Accuracy**	**Langugage**
KIS-1	0.6286	Japanese		UA-LM	0.717	Japanese
KIS-2	0.6286	Japanese		UA-TFIDF	0.692	Japanese
iLis7	**0.6286***	English		KIS-YN-S	0.653	Japanese
KIS-3	0.5857	Japanese		NAIST2	0.653	Japanese
KIS-4	0.5857	Japanese		NAIST1	0.615	Japanese
N01-5	0.5714	English		iLis9-1	**0.576***	English
UofA	0.5571	Japanese		iLis7	0.564	English
Our models						
CBOW	0.4595	English		CBOW	0.5256	English
BiLSTM	**0.6757**	**English**		BiLSTM	**0.6410**	English
Decomposable	0.6081	English		Decomposable	0.5897	English
ESIM	0.5541	English		ESIM	**0.6410**	English

Besides, the results on Dataset 1 show that models which are trained on small datasets are not stable. The overall result of the binary classification task is around 50%. In addition, in some cases, the predictions of these models are biased into the majority class based on the development set. For example, the model, which is trained with ESIM in

Dataset 1, predicts 'NO' for all instances in the test set. In these cases, $avgF_1$ measure is more suitable than *Accuracy* because $avgF_1$ is very low if the model biases into a class.

In four different deep learning models, BiLSTM model produces the best results on both of two datasets. BiLSTM is not simple as CBOW and is not complicated as ESIM and the decomposable attention model. In our opinion, although the size of our dataset has been increased, it is still very small in comparison with other datasets such as SNLI [Bowman et al., 2015]. Therefore, sometimes, complex models such as ESIM do not produce the best results. Although CBOW models trained on Dataset 3 achieved quite good results, the performance in Dataset 2 quite poor. In these approaches, the sentence encoding-based model with BiLSTM is a good choice for this task.

The average performances in two augmented datasets (dataset 2 and 3) are quite comparable. Dataset 3 is constructed from analyzing R-E structures in which R-E parts are automatically recognized using the pre-trained RE parser. It demonstrates that the quality of RE parser tool quite good.

We also conduct experiments to evaluate the contribution of each generated dataset. Table 4.8 shows the experimental results of different dataset combinations. The model trained on Dataset 2 which is the combination of the original datasets and newly instances generated from both syntactic parse trees and R-E structures show the best performance. However, the performance decreases when we only combine the original dataset with a single generated dataset (see experimental result on Dataset 4,5 and 6). In different models trained on different augmented datasets, BiLSTM shows the best performance almost cases. The model trained with BiLSTM on Dataset 2 shows the highest performance with the low standard deviation.

In comparison with previous works, our best system outperforms than other previous approaches on the COLIEE English dataset (see Table 4.9). In comparison with best systems in COLIEE 2016 and 2017, the performance of BiLSTM improves from 5% to 7%.

Table 4.10 shows the output of models which are trained on three datasets for 4 input pairs. The results show that our models can capture some simple phenomena as negation and sub-sentence phenomena which are also popular in entailment task in legal texts.

4.6.3 Experimental Results of Multi-Sentence Entailment Classification Model

Model settings: We choose word embedding size $d = 100$, hidden layer size in MLP = 100, batch size = 32, drop-out rate = 0.2. All models are trained with Stochastic Gradient Descent optimizer [Kiefer and Wolfowitz, 1952] with the learning rate of 0.01. Each model is trained within 60 epochs, and we choose the one that produces the best performance on the development set. Word embeddings are also initialized in the same way as models in the previous section. In this study, we only use the itemization detection method to decompose long sentences into list of simple sentences.

Results: Table 4.11 shows experimental results of different Multi-Sentence models and comparisons with previous approaches. Compared to previous models, the proposed model produces comparable results. In two encoding methods in Multi-Sentence, although BiLSTM seems not to be effective, Multi-Sentence-CBOW achieves the best result in the test set H28 and a quite good result on H27.

Table 4.10: Sample output of our systems on different models trained on different datasets

Pair	Content of artcile and question	Gold	Model	Dataset 1	Dataset 2
1	**t:** The establishment of a juridical person may **not** be asserted against a third party . **h:**The establishment of a juridical person may be asserted against a third party .	NO	CBOW BI-LSTM ESIM Decom.	**YES** NO NO NO	**NO** **NO** **NO** **NO**
2	**t:** The establishment of a juridical person may be asserted against a third party . **h:**The establishment of a juridical person may **not** be asserted against a third party .	NO	CBOW BI-LSTM ESIM Decom.	**YES** NO NO NO	**NO** **NO** **NO** **NO**
3	**t:** A mandate may be cancelled by either party at any time . **h:** A mandate may be cancelled .	YES	CBOW BI-LSTM ESIM Decom.	**YES** NO NO NO	NO **YES** **YES** **YES**
4	**t:** If parents divorce by agreement , they may agree upon which parent shall have parental authority in relation to a child . **h:**In cases where a child between a couple during marriage is a minor , if parents divorce by agreement , they may agree upon which parent shall have parental authority in relation to a child without obtaining the permission of the family court . (question H27-29-U in COLIEE data set)	YES	CBOW BI-LSTM ESIM Decom.	**YES** NO NO YES	NO **YES** **YES** **YES**

Table 4.12 shows the contributions of sentence decomposition. In case of sentence decomposition is not used, related articles are decomposed into sentences by sentence splitting. The average results show that the sentence decomposition yield positive effects. The average performance improves by 2% if the sentence decomposition step is used.

Table 4.11: Comparison between Multi-Sentence Models (5-6) and Single-Sentence models (1-4). All models using the same setting and all scores are the accuracy score of classifiers. Symbols * indicate the best result in each test set.

Model	H27	H28
Sentence encoding-based models		
(1) CBOW	0.4595	0.5256
(2) BiLSTM	**0.6757***	0.641
Attention-based models		
(3) Decomposable attention model	0.6081	0.5897
(4) ESIM	0.5541	0.641
Multi-Sentence Model		
(5) Multi-Sentence-CBOW	0.5946	**0.6795***
(6) Multi-Sentence-BiLSTM	0.527	0.5513

Table 4.12: Comparison between Sentence Decomposition and Normal Sentence Splitting. Experimental result of Multi-Sentence-CBOW. All scores are the accuracy score of classifiers

Batch Norm	**Test set**	**# Question**	**No Decomposing**	**Long Sentence Decomposing**
	H27	74	0.5541	0.5541
	H28	78	0.5000	0.5769
x	H27	74	0.6351	0.5946
x	H28	78	0.6410	0.6795
Average			**0.5822**	**0.6020**

4.7 Conclusions and Future Work

In this study, we first propose a semi-supervised approach which employs an unsupervised method to generate weak labeled data; this augmented data set then is combined with the original dataset to train entailment classifiers using supervised learning algorithms. The data augmentation methods are based on analyses of syntactic and logical structures of legal sentences. We then apply several deep learning-based models for RTE in legal texts including sentence encoding-based methods and attention-based methods. We also propose a method to decompose related articles in the input pair into a list of simple sentences. We next propose a novel entailment classification model that can handle related articles as a list of simple sentences instead of a very long sentence. Experimental results on two official test sets demonstrate that the augmented datasets exhibit positive effects in the COLIEE's entailment classification task. The performance of Multi-Sentence model is also comparable to previous best performance.

In future, incorporating new features or generating new data that covers other linguistic phenomena for this task are ways to improve the performance.

Chapter 5

Applications in Question Answering Systems

This chapter presents an application of the RTE component in a question answering system.

5.1 Introduction

In the age of "big data", the task of finding information to answer some questions by hand is too expensive and complex. With the aids of computers, people usually do these tasks by using an Information Retrieval system or a Question Answering system. Compared to an information retrieval system, a question answering system is at a higher level because it not only retrieves relevant documents but can answer questions from users. Information retrieval systems usually use the keyword-based approach to search collections of documents that are similar to the given query. The users then examine individual documents in the results list to identify and extract the information they need. Question Answering systems provide more benefits. It can directly answer questions from users and it can provide documents that support the answers.

Most current question answering systems focus on factoid questions. Factoid questions are questions that can be answered with simple facts and answers are usually expressed in short texts.

Two main paradigms of question answering systems for answering factoid questions are IR-based question answering and knowledge-based question [Jurafsky and Martin, 2009]. IR-based question answering relies on the enormous amounts of information available as text on the Web or in collections of specific domains such as PubMed [1]. Given a user question, an IR-based question answering system will answer the question by executing several steps such as Question Processing, Passage Retrieval, Answer Processing. A typical architecture of an IR-based factoid question answering system is illustrated in Figure 5.1 and a factoid question answering system is described in Bian et al. [2008]. The second paradigm is knowledge-based question answering. In this paradigm, a query will be converted into a semantic representation, and the answer is extracted by querying in a database of facts such as Freebase [Bollacker et al., 2008] or DBpedia [Auer et al.,

[1] PubMed is a free full-text archive of biomedical and life sciences journal literature at the U.S. National Institutes of Health's National Library of Medicine (NIH/NLM)

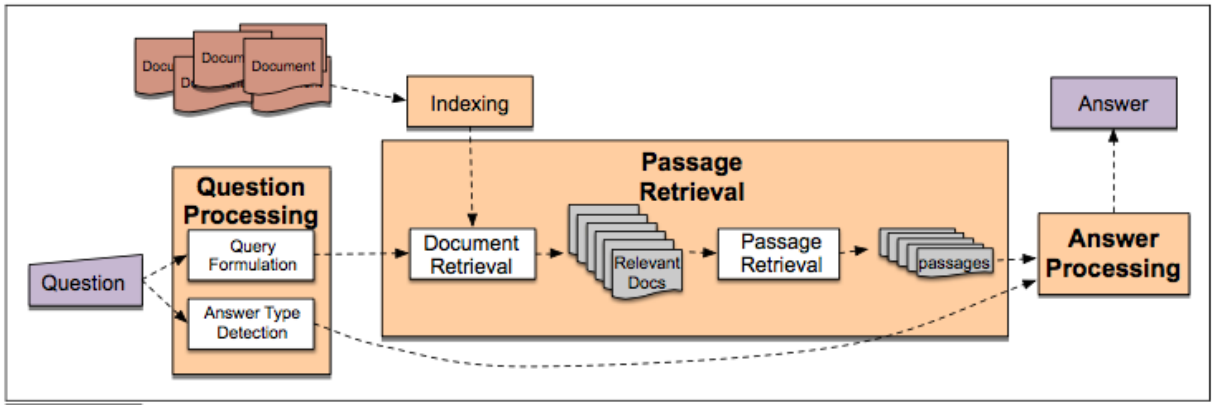


Figure 5.1: The typical architecture of an IR-based factoid question answering systems [Jurafsky and Martin, 2009]

2007]. Other question answering systems use the hybrid approach such as IBM's Watson [Ferrucci, 2011].

There are a few of research for building question answering systems for the legal domain such as [Quaresma and Rodrigues, 2005, Monroy et al., 2009, Bennett et al., 2017]. For example, Quaresma and Rodrigues [2005] has proposed an architecture for a question answering system for the Portuguese language and we applied it to the legal domain which is a knowledge-based question answering system. The system first, uses NLP techniques to create a knowledge base with the information conveyed by documents. Queries then are analyzed by the same tools. Finally, logical inferences over the knowledge base are performed to find an answer. Monroy et al. [2009] has proposed a question answering system for Spanish at the shallow level by using graphs. The system can output relevant articles to the question based on the similarity.

In this study, we focus on building a question answering system for the legal domain follow the aims of Competition on Legal Information Extraction/Entailment (COLIEE). The system can answer Yes/No question in the Japanese Legal Bar exams. Compared to question answering datasets such as SQUAD [Rajpurkar et al., 2016] and TriviaQA [Joshi et al., 2017] which questions are indicated by Wh-words followed by a topic, questions in COLIEE dataset are statements which need to answer Yes/No. Answering questions in SQUAD and Trivia datasets seems simpler than COLIEE because the question word and the question type usually reveal the clues for finding answers (e.g. What genre, Who is, Where). We need to find the correct text spans in the reference text to extract the answers based on these clues. However, in COLIEE, although the answer is limited in *Yes* or *No*, we need to analyze the semantics of the given statement and the whole content of its related articles deeply to decide whether or not the given statement is correct. Table 5.1 shows some examples of questions in SQUAD, TriviaQA and COLIEE datasets.

We build a two-phase system for the legal information question answering task. In the first phase, a list of relevant articles is retrieved by computing the cosine similarity between the TF-IDF vectors of the given question and articles. The architecture of this phase follows the architecture described in [Nguyen et al., 2016c] with some improvements in the indexing step. In particular, we apply the n-gram words indexing model beside the uni-gram word indexing model. Experimental results show that our approaches have some promising results. First, adding bi-gram and tri-gram indexing models shows a significant

Dataset	Examples
SQUAD	- What causes precipitation to fall? - Where do water droplets collide with ice crystals to form precipitation?
TriviaQA	- Who won the Nobel Peace Prize in 2009? - Which politician won the Nobel Peace Prize in 2009?
COLIEE	- H18-1-1: A special provision that releases warranty can be made, but in that situation, when there are rights that the seller establishes on his/her own for a third party, the seller is not released of warranty. - H18-1-2: In cases where a person plans to prevent crime in their own house by fixing the fence of a neighboring house, that person is found as having intent towards the other person.

Table 5.1: Questions in different QA dataset

improvement.

In the second phase, we employ the pre-trained models for finding a Yes/No answer given a user's question. Finally, the relevant articles and the answer will be displayed to the user via a web interface. The details of the architecture and the implementation of our system are presented in next sections.

5.2 System Architecture

The architecture of our question answering system has two phases including *Relevant Analysis* and *Legal Question Answering*. Given a question from users, the system first retrieves relevant articles based on the similarity between the vector represented for the question and vectors which presented for articles in the corpus which is a set of legal articles. Then the pair of the given question and its most relevant article are passed through the *Legal Question Answering* phase to classify whether or not the given question is entailed from its relevant article. Then, relevant articles and the answer are displayed to end users. A scenario of this question answering system is shown in Figure 5.2 and the architecture the system is illustrated in Figure 5.3.

5.2.1 Relevant Analysis

Indexing: This phase will convert all articles into TF-IDF vector representations, a popular method for representing documents in the information retrieval field [Manning et al., 2008]. Moreover, to improve the performance of the system, some pre-processing steps can be applied such as stemming or removing stop words. Besides, to increase the importance of long text matching, instead of using only uni-gram model, we add bi-gram and tri-gram indexing model for documents representation.

Query Expansion: The query expansion step is an option in our system. This step tries to add related terms into a given query to improve retrieve performance. Query expansion

Figure 5.2: The example of of end-to-end Question Answering System

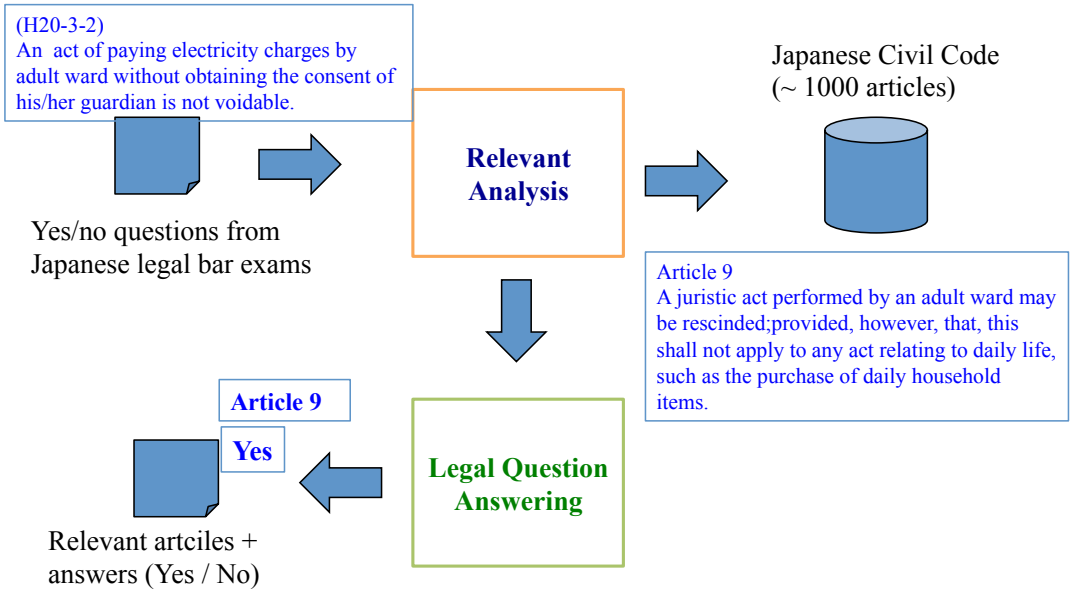


Figure 5.3: The architecture of end-to-end Question Answering System

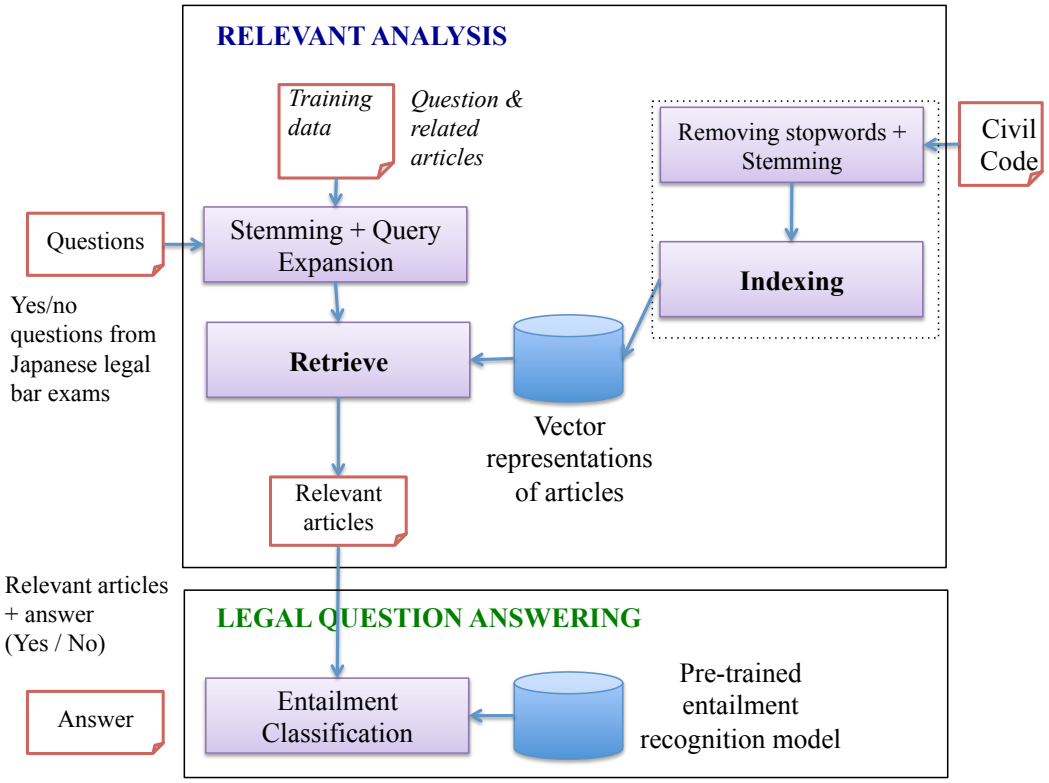


Table 5.2: An example of query expansion using **word2vec**

Query (H18-1-3)	A compulsory **auction** is also a sale, so warranty is imposed the same as for an ordinary **sale**.
Relevant Article (Article 568)	(Warranty in cases of Compulsory Auctions) (1)The successful **bidder** at compulsory auction may cancel the contract or demand a reduction from the **purchase** money against the obligor in accordance with the provisions from Article 561 through to the preceding Article. ...
Pairs added to the dictionary	(sale, purchase), (auction, bidder)

involves techniques that can find synonyms of words, and search for the synonyms as well. We employed two methods for query expansion using *Word2Vec* and *WordNet*:

- Query expansion using *Word2Vec*[Mikolov et al., 2013]: From questions and their relevant articles in training corpus, we extract similar word pairs of a word in question and articles base on cosine similarity of their word embedding representation. We select word-pairs that have the cosine similarly value ≥ 0.5 as the related words for query expansion. We then create a dictionary of similar pairs then this dictionary will be used for query expansion for a new query. This method is expected to retrieve articles that do not share words with the given query. We also remove all stop words before extracting similar word pairs.
- Query expansion using *WordNet* [Miller, 1995]: From questions, we expand the question by adding synonyms and hypernyms of words in that question by looking in WordNet dictionary. However, this way is not effective because each word may have many senses and we do not know the sense of that word in the question. Consequently, this method adds many un-related words, so the precision reduces sharply.

Retrieve: To retrieve relevant articles for a given question, we select the article that has the highest similarity score with the question. The similarity score of a question and an article is computed based on the cosine similarity between two vector representations of the question and the article.

$$\textbf{similarity}(\text{question}, \text{article}) = cosine(q, a) \tag{5.1}$$

where q, a are vector representations of the question and the article

5.2.2 Legal Question Answering

After the most relevant article has been identified by *Relevant Analysis* step, an RTE classifier trained from Chapter 3 is employed to classify whether or not this article entails the given question. Finally, the output including the most related article and the answer of the RTE classifier are displayed to end users via a web interface.

5.3 Experiments and Results

5.3.1 Relevant Analysis

For phase one, we analyzed aspects of the information retrieval task in our system including stemming, query expansion, n-gram indexing, removing stop-words, and ranking. We conducted experiments and compared different configurations to find the configuration that produces the best performance. Table 5.3 shows the results of the information retrieval phase.

The results on the H18-H25 show the contribution of the stemming and removing stop words step. When we stem or remove stop words, the performance of the system is usually better. However, these steps have the negative impact on the data set H26 and H28.

Adding bi-gram indexing and tri-gram indexing models also improve the performance of our system. In all three data sets, top 3 best results are always using bi-gram or tri-gram indexing model. Experimental results in Table 5.4 show that adding 2-gram and 3-gram have an important contribution for the retrieval task. The performance improves significantly when 2-gram and 3-gram indexing models are used. For example, the retrieval performance improves 1.93%, 4.42%, 7.44% on H18-H25, H26, H28 data sets if we use the 2-gram indexing model. However, when we use the n-gram indexing model with $n > 3$, the results do not improve but it takes more time for retrieving as well as for indexing.

The query expansion step does not always improve the results because using word embedding similarity can find useful terms to the given question for but it may add many unrelated terms.

5.3.2 Entailment classification

The evaluation of entailment classification models presented in Chapter 4 is conducted with the assumption that correct relevant articles have been provided. However, our system consists of two phases, so the performance at the second phase (Legal Question Answering) may be affected by the performance of the first phase. For example, if the performance of *Relevant Analysis* step is low, the system may not found the correct relevant articles of a given question. Consequently, *Legal Question Answering* may answer the question incorrectly. Table 5.5 shows the evaluation of our question answering system on test sets H27 and H28. In general, the performance of Phase 2 is somehow affected by the performance of Phase 1.

Table 5.6 shows outputs of our system for a question in the test set H28 in which the article retrieved by Relevant Analysis phase is correct. In the Legal Question Answering phase, models trained on the dataset 3 (our generated dataset) seem to be better than models trained on dataset 1 (original dataset).

5.4 Conclusions and Future Work

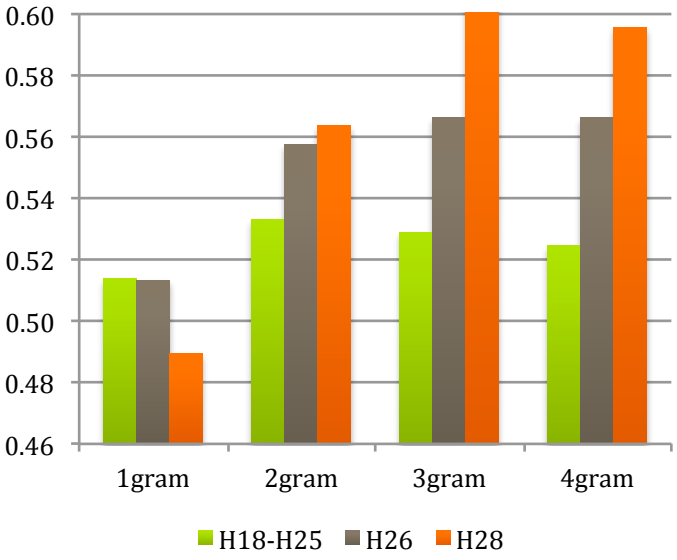
In this study, we propose an architecture for an end-to-end question answering system which can answer Yes/No questions related to Japanese Civil Code (English version). The first phase is a traditional IR component which retrieves relevant articles. Beside some popular methods for questions and documents processing such as stemming, removing stop-words, query expansion, we employ an n-gram word indexing which exhibits

Table 5.3: Experimental results ($F_{\beta=1}$ score) of phase 1 - Relevant Analysis. (a) indicates the performance is the best on the test set but training set and development set; (b), (c) are the results on the test set when the performance is the best on the development set (H26) and training set (H18-H25) respectively.

#	QUERY-EXPANSION	N-GRAM	REMOVE-STOPWORD	STEM	H18-H25	H26	H28
1		1gram			0.5096	0.5752	0.5319
2		1gram		x	0.5203	0.5487	0.5000
3		1gram	x		0.5075	0.5221	0.5532
4		1gram	x	x	0.5139	0.5133	0.4894
5		2gram			0.4989	0.5398	0.5426
6		2gram		x	0.5139	0.5487	0.5319
7		2gram	x		0.5096	0.5487	0.5638
8		2gram	x	x	0.5332	0.5575	0.5638
9		3gram			0.4968	0.5575	0.5532
10		3gram		x	0.5139	0.5752	0.5426
11		3gram	x		0.5161	0.5841	0.5638
12		3gram	x	x	0.5289	0.5664	**0.6277**$^{(a)}$
13		4gram			0.4946	**0.6018**	**0.5638**$^{(b)}$
14		4gram		x	0.5032	0.5929	0.5319
15		4gram	x		0.5139	0.5664	0.5638
16		4gram	x	x	0.5246	0.5664	0.5957
17	x	1gram			0.5203	0.5752	0.5213
18	x	1gram		x	0.5118	0.5310	0.5106
19	x	1gram	x		0.5139	0.5133	0.5426
20	x	1gram	x	x	0.5032	0.4956	0.5213
21	x	2gram			0.5075	0.5487	0.5745
22	x	2gram		x	0.5310	0.5487	0.5319
23	x	2gram	x		0.5246	0.5487	0.5532
24	x	2gram	x	x	**0.5439**	0.5310	**0.5851**$^{(c)}$
25	x	3gram			0.5032	0.5664	0.5532
26	x	3gram		x	0.5203	0.5752	0.5532
27	x	3gram	x		0.5246	0.5664	0.5532
28	x	3gram	x	x	0.5353	0.5664	0.5957
29	x	4gram			0.5011	0.5841	0.5638
30	x	4gram		x	0.5075	0.5841	0.5319
31	x	4gram	x		0.5268	0.5664	0.5426
32	x	4gram	x	x	0.5268	0.5575	0.5957

Table 5.4: Comparison between difference n-gram indexing models (all other configurations are the same: Query Expansion:No, Remove Stop words: Yes, Stemming: Yes)

#	N-GRAM	H18-H25	H26	H28
4	1 gram	0.5139	0.5133	0.4894
8	2 gram	0.5332	0.5575	0.5638
12	3 gram	0.5289	0.5664	0.6277
16	4 gram	0.5246	0.5664	0.5957



significant improvements for relevant analysis. The Legal Question Answering phase will answer the question by employ pre-trained models of our study presented in Chapter 4 to classify whether or not the question is entailed from its most relevant article. Our system is the winner of the Information Retrieval task for the live competition of COLIEE 2017. Currently, the system does not use any deep analysis of questions and articles. In future, analyzing questions and articles deeply is a way to improve the quality of our question answering.

Table 5.5: Performance of RTE classifiers on test sets H27 and H28. The end-to-end evaluation is conducted after relevant articles are retrieved from Relevant Analysis phase (Configuration: Indexing-model:3gram; Stemming:yes; Remove stop words:yes); Evaluation only for phase 2 is conducted with the assumption that relevant articles have been provided)

Evaluation on H27

Performance of Relevant Analysis step: 0.6622,R=0.4537;F=0.5385

Model	**End-to-end evaluation**	**Evaluation only for phase 2**
BiLSTM	0.6081	0.6757
CBOW	0.4865	0.4869
Decomposable Att.	0.6216	0.6081
ESIM	0.5	0.5541

Evaluation on H28

Performance of Relevant Analysis step:P=0.7564,R=0.5364, F=0.6277

Model	**End-to-end evaluation**	**Evaluation only for phase 2**
BiLSTM	0.5897	0.6410
CBOW	0.6154	0.6154
Decomposable Att.	0.5128	0.6026
ESIM	0.5256	0.6410

Table 5.6: An output for an question in the test set of our system

QUESTION	**Relevant Articles (GOLD)**	**Entailment Label**
H28-28-1: A person who has tendered anything as performance of an obligation may not demand the return of the thing tendered if the person were negligent in not knowing that the obligation did not exist.	Article 705	**NO**

SYSTEM OUTPUT			**CORRECT (x)**
Relevant Analysis	Article 705 (Performance knowing of Absence of Obligation) (similarity score= 0.750977641732) A person who has tendered anything as performance of an obligation may not demand the return of the thing tendered if the person knew , at the time , that the obligation did not exist .		x
Legal Question Answering	BiLSTM (trained on dataset 1)	YES	o
	CBOW (trained on dataset 1)	YES	o
	Decomposable Att. (trained on dataset 1)	YES	o
	ESIM (trained on dataset 1)	NO	x
	BiLSTM (trained on dataset 2)	NO	x
	CBOW (trained on dataset 2)	NO	x
	Decomposable Att. (trained on dataset 2)	NO	x
	ESIM (trained on dataset 2)	YES	o

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Our thesis is motivated by the fact that legal texts analysis and textual entailment recognition will benefit for many applications in the legal domain, and deep learning are a promising approach for solving these tasks.

The main contributions of this dissertation are summarized as follows:

- We propose several deep learning-based models for recognizing requisite-effectuation parts in legal texts (Chapter 3). We first propose the BiLSTM-CRF model which allows using external features such as Part-of-Speech and several syntactic-based features. We then propose several approaches for recognizing overlapped RE parts including the sequence of Bi-LSTM-CRF for the cascading approach propose and two novel models called Multilayer-BiLSTM-CRF and Multilayer-BiLSTM-MLP-CRF for the unified model approach. The proposed approaches exhibit significant improvements compared to previous approaches. We also deploy pre-trained RRE passers as services that can be called by third-party applications.

- We propose two methods for data augmentation which can improve the performance of RTE on the COLIEE entailment task (Chapter 4). These methods are based on the analysis of requisite-effectuation structures and syntactic parse tree of legal sentences. We also apply several deep learning models for recognizing textual entailment in legal texts. Besides, we propose some methods for decomposing a long legal sentence into a list of simple sentences such as analyzing itemization expressions in legal sentences and analyzing R-E structures of legal sentences. We then propose a novel deep learning model for RTE that can handle multiple sentences instead of a single sentence.

- We finally present an application of RTE for building a question answering system for the legal domain (Chapter 5). The system can answer yes/no questions in Japanese Civil Code. This is the first attempt to build such systems and there are a lot of changes to improve it in future.

6.2 Future Work

The next study will focus on the following things:

- **Legal text processing in other languages**: All proposed approaches in Chapter 3 and Chapter 4 are deep learning-based models that do not need a strong engine for feature engineering. Besides, the design of our models is very extensible to solve a general sequence labeling problem. For example, it is simply to add features or increase the number of layers into a BiLSTM-CRF-F and Multi-BiLSTM-MLP-CRF). Therefore, these approaches can be applied for analyzing structures and recognizing textual entailment in legal texts of another language easily. For example, we can apply these models to extend the studies of [Nguyen et al., 2015] and [Nguyen et al., 2016a] which are first attempts to analyze logical parts in Vietnamese legal texts. These models can be applied to analyze other components of legal texts by modeling it as a sequence labeling task.

- **Applying these proposed models to other tasks in NLP**: The proposed models in Chapter 3 are designed for labeling sequential data. It can be applied to other tasks in language processing such as named entity recognition, information extraction, semantic role labeling, shallow discourse parsing in both of general and specific domain such as scientific papers and bio-medical texts. For example, in shallow discourse parsing task [1], we can apply the multilayer models to recognize arguments of a discourse relation by treating this task as a sequence labeling. We can then apply entailment classification models in Chapter 4 to classify the relationship between two identified arguments.

- **Studying semi-supervised methods and feature engineering methods for RTE task**: The COLIEE dataset used in our study still small. In future, applying other methods to generate weak labeled data and incorporating knowledge from different source domain or extracting features by analyzing legal texts deeply are ways to improve the performance of RTE task. Besides, we can recognize the entailment between two long texts by decomposing them into small parts in which the RTE problem can be solved easier.

- **Building information retrieval and question answering systems in legal domain**: With the proposed models, we would like to build information retrieval system in the legal domain which can retrieve legal articles. Legal articles firstly can be analyzed to extract requisite-effectuation parts from an RE parser. Queries from users can be searched in different regions of articles which may show more benefits to users. Besides, we can build a legal question answering system in which RRE and RTE components are important components.

[1] `http://www.cs.brandeis.edu/~clp/conll15st/intro.html`

Bibliography

K. J. Adebayo, L. D. Caro, G. Boella, and C. Bartolini. An approach to information retrieval and question answering in the legal domain. In *The 10th International Workshop on Juris-Informatics (JURISIN)*, 2016.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

R. Bar Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The second pascal recognising textual entailment challenge. 2006.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Z. Bennett, T. Russell-Rose, and K. Farmer. A scalable approach to legal question answering. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 269–270. ACM, 2017.

L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo. The fifth pascal recognizing textual entailment challenge.

J. Bian, Y. Liu, E. Agichtein, and H. Zha. Finding the right facts in the crowd: factoid question answering over social media. In *Proceedings of the 17th international conference on World Wide Web*, pages 467–476. ACM, 2008.

M. Boden. A guide to recurrent neural networks and backpropagation. 2001.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, and C. Potts. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*, 2016.

Q. Chen, X. Zhu, Z. Ling, S. Wei, and H. Jiang. Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

J. Cheng, L. Dong, and M. Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

J. P. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*, 2015.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12 (Aug):2493–2537, 2011.

I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pages 177–190, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33427-0, 978-3-540-33427-9. doi: 10.1007/11736790_9. URL `http://dx.doi.org/10.1007/11736790_9`.

I. Dagan, B. Dolan, B. Magnini, and D. Roth. The fourth pascal recognizing textual entailment challenge. *Journal of Natural Language Engineering*, 2010.

I. Dagan, D. Roth, M. Sammons, and F. M. Zanzotto. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4): 1–220, 2013.

P.-K. Do, H.-T. Nguyen, C.-X. Tran, M.-T. Nguyen, and M.-L. Nguyen. Legal question answering using ranking svm and deep convolutional neural network. 2016.

C. Dozier, R. Kondadadi, M. Light, A. Vachher, S. Veeramachaneni, and R. Wudali. *Named Entity Recognition and Resolution in Legal Text*, pages 27–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-12837-0. doi: 10.1007/978-3-642-12837-0_2. URL `https://doi.org/10.1007/978-3-642-12837-0_2`.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

D. A. Ferrucci. Ibm's watson/deepqa. In *ACM SIGARCH Computer Architecture News*, volume 39. ACM, 2011.

G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

M. A. R. Gaona, A. Gelbukh, and S. Bandyopadhyay. Recognizing textual entailment using a machine learning approach. In *Mexican International Conference on Artificial Intelligence*, pages 177–185. Springer, 2010.

A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 2017.

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

Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

D. Jurafsky and J. H. Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2009.

Y. Kano, R. Hoshino, and R. Taniguchi. Analyzable legal yes/no question answering system using linguistic structures. In K. Satoh, M.-Y. Kim, Y. Kano, R. Goebel, and T. Oliveira, editors, *COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment*, volume 47 of *EPiC Series in Computing*, pages 57–67. EasyChair, 2017a.

Y. Kano, M.-Y. Kim, R. Goebel, and K. Satoh. Overview of coliee 2017. In K. Satoh, M.-Y. Kim, Y. Kano, R. Goebel, and T. Oliveira, editors, *COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment*, volume 47 of *EPiC Series in Computing*, pages 1–8. EasyChair, 2017b.

A. Karpathy. The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*, 2015.

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

J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.

K. Kim, S. Heo, S. Jung, K. Hong, and Y.-Y. Rhim. Ensemble based legal information retrieval and entailment system. In *The 10th International Workshop on Juris-Informatics (JURISIN)*, 2016a.

M.-Y. Kim and K. Goebel, Randy Satoh. Coliee-2015 : Evaluation of legal question answering. In *Ninth International Workshop on Juris-informatics (JURISIN)*, 2015.

M.-Y. Kim and R. Goebel. Two-step cascaded textual entailment for legal bar exam question answering. In *COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment*.

M.-Y. Kim, R. Goebel, Y. Kano, and K. Satoh. Coliee-2016: Evaluation of the competition on legal information extraction and entailment. 11 2016b.

M.-Y. Kim, Y. Xu, Y. Lu, and R. Goebel. Legal question answering using paraphrasing and entailment analysis. In *The 10th International Workshop on Juris-Informatics (JURISIN)*, 2016c.

Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

T. Kudo. Crf++: Yet another crf toolkit. *Software available at http://crfpp. sourceforge. net*, 2005.

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

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

R. Leaman and G. Gonzalez. Banner: an executable survey of advances in biomedical named entity recognition. In *Biocomputing 2008*, pages 652–663. World Scientific, 2008.

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

Y. Liu, C. Sun, L. Lin, and X. Wang. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*, 2016.

M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

P. Malakasiotis and I. Androutsopoulos. Learning textual entailment using svms and string similarity measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 42–47. Association for Computational Linguistics, 2007.

C. D. Manning, P. Raghavan, and H. Schütze. Scoring, term weighting and the vector space model. *Introduction to information retrieval*, 100:2–4, 2008.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38 (11):39–41, 1995.

A. Monroy, H. Calvo, and A. Gelbukh. Nlp for shallow question answering of legal documents using graphs. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 498–508. Springer, 2009.

A. Morimoto, D. Kubo, M. Sato, H. Shindo, and Y. Matsumoto. Legal question answering system using neural attention. In K. Satoh, M.-Y. Kim, Y. Kano, R. Goebel, and T. Oliveira, editors, *COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment*, volume 47 of *EPiC Series in Computing*, pages 79–89. EasyChair, 2017.

L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin. Natural language inference by tree-based convolution and heuristic matching. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 130, 2016.

T. Munkhdalai and H. Yu. Neural semantic encoders. *CoRR*, abs/1607.04315, 2016a. URL `http://arxiv.org/abs/1607.04315`.

T. Munkhdalai and H. Yu. Neural tree indexers for text understanding. *CoRR*, abs/1607.04492, 2016b. URL `http://arxiv.org/abs/1607.04492`.

M. Nakamura, S. Nobuoka, and A. Shimazu. Towards translation of legal sentences into logical forms. In *Annual Conference of the Japanese Society for Artificial Intelligence*, pages 349–362. Springer, 2007.

X. B. Ngo, L. M. Nguyen, and A. Shimazu. Recognition of requisite part and effectuation part in law sentences. In *Proceedings of (ICCPOL)*, pages 29–34, 2010.

X. B. Ngo, L. M. Nguyen, T. O. Tran, and A. Shimazu. A two-phase framework for learning logical structures of paragraphs in legal articles. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12(1):3, 2013.

L.-M. Nguyen, N. X. Bach, and A. Shimazu. 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*, pages 21–29. Association for Computational Linguistics, 2011.

T. S. Nguyen, T. D. Nguyen, B. Q. Ho, and L. M. Nguyen. Recognizing logical parts in vietnamese legal texts using conditional random fields. In *Computing & Communication Technologies-Research, Innovation, and Vision for the Future (RIVF), 2015 IEEE RIVF International Conference on*, pages 1–6. IEEE, 2015.

T. S. Nguyen, L. M. Nguyen, B. Q. Ho, and A. Shimazu. Recognizing logical parts in legal texts using neural architectures. In *Knowledge and Systems Engineering (KSE), 2016 Eighth International Conference on*, pages 252–257. IEEE, 2016a.

T. S. Nguyen, L. M. Nguyen, and X. C. Tran. Vietnamese named entity recognition at vlsp 2016 evaluation campaign. In *In Proceedings of The Fourth International Workshop on Vietnamese Language and Speech Processing*, pages 18–23, 2016b.

T.-S. Nguyen, V.-A. Phan, T.-H. Nguyen, H.-L. Trieu, N.-P. Chau, T.-T. Pham, and L.-M. Nguyen. Legal information extraction/entailment using svm-ranking and tree-based convolutional neural network. In *The 10th International Workshop on Juris-Informatics (JURISIN)*, 2016c.

T.-S. Nguyen, V.-A. Phan, and L.-M. Nguyen. Recognizing entailments in legal texts using sentence encoding-based and decomposable attention models. In K. Satoh, M.-Y. Kim, Y. Kano, R. Goebel, and T. Oliveira, editors, *COLIEE 2017. 4th Competition on Legal Information Extraction and Entailment*, volume 47 of *EPiC Series in Computing*, pages 31–42. EasyChair, 2017.

B. Paria, K. Annervaz, A. Dukkipati, A. Chatterjee, and S. Podder. A neural architecture mimicking humans end-to-end for natural language inference. *arXiv preprint arXiv:1611.04741*, 2016.

A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

F. Peng and A. McCallum. Information extraction from research papers using conditional random fields. *Information processing & management*, 42(4):963–979, 2006.

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

P. Quaresma and I. P. Rodrigues. A question answer system for legal information retrieval. In *JURIX*, pages 91–100, 2005.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

T. Rocktäschel, M. Weidlich, and U. Leser. Chemspot: a hybrid system for chemical named entity recognition. *Bioinformatics*, 28(12):1633–1640, 2012.

T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.

B. Settles. 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*, pages 104–107. Association for Computational Linguistics, 2004.

F. Sha and F. Pereira. 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*, pages 134–141. Association for Computational Linguistics, 2003.

L. Sha, B. Chang, Z. Sui, and S. Li. Reading and thinking: Re-read lstm unit for textual entailment recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2870–2879, 2016.

A. Shimazu. Structural paraphrase of law paragraphs. In *Eleventh International Workshop on Juris-informatics (JURISIN)*, 2017.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

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

Y. M. Taku Kudo. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

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

R. Taniguchi and Y. Kano. *Legal Yes/No Question Answering System Using Case-Role Analysis*, pages 284–298. Springer International Publishing, Cham, 2017. ISBN 978-3-319-61572-1. doi: 10.1007/978-3-319-61572-1_19. URL `https://doi.org/10.1007/978-3-319-61572-1_19`.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

P. Wang, Y. Qian, F. Soong, L. He, and H. Zhao. A unified tagging solution: Bidirectional lstm recurrent neural network with word embedding. *arXiv preprint arXiv:1511.00215*, 2015a.

P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *arXiv preprint arXiv:1510.06168*, 2015b.

S. Wang and J. Jiang. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*, 2015.

F. M. Zanzotto, M. Pennacchiotti, and A. Moschitti. A machine learning approach to textual entailment recognition. *Natural Language Engineering*, 15(4):551–582, 2009.

M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

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

Publications and Awards

Journals

[1] Truong-Son Nguyen, Le-Minh Nguyen, Ken Satoh, Satoshi Tojo and Akira Shimazu: "Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts," Artificial Intelligent and Law, Volume 26, Issue 2, pages 169–199, 2018 (DOI: `https://doi.org/10.1007/s10506-018-9225-1`).

Conference papers

[2] Truong-Son Nguyen, Le Minh Nguyen, and Ken Satoh: "Improving entailment recognition in legal texts using corpus generation," in Proceedings of Second International Workshop on SCIentific DOCument Analysis (SCIDOCA), 2017.

[3] Truong-Son Nguyen, Le-Minh Nguyen, Akira Shimazu and Kiyoaki Shirai: "Structural Paraphrasing in Japanese Legal Texts ", Eleventh International Workshop on Juris-informatics (JURISIN), pages 62–75, 2017.

[4] Truong-Son Nguyen, Le-Minh Nguyen, Ken Satoh, Satoshi Tojo and Akira Shimazu: "Single and multiple layer BI-LSTM-CRF for recognizing requisite and effectuation parts in legal texts", In Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts (ASAIL), 2017.

[5] Truong-Son Nguyen, Viet-Anh Phan, Le-Minh Nguyen: "Recognizing entailments in legal texts using sentence encoding-based and decomposable attention models", In Proceedings of 4th Competition on Legal Information Extraction and Entailment (COLIEE), pages 31–42, 2017.

[6] Truong-Son Nguyen, and Le-Minh Nguyen: "Nested named entity recognition using multilayer recurrent neural networks", In Proceedings of Conference of the Pacific Association for Computational Linguistics (PACLING), pages 233–246, 2017.

[7] Dac-Viet Lai, Truong-Son Nguyen, and Le-Minh Nguyen: "Deletion-based sentence compression using Bi-enc-dec LSTM", In Proceedings of Conference of the Pacific Association for Computational Linguistics (PACLING), pages 249–260, 2017.

[8] Truong-Son Nguyen, Le Minh Nguyen, and Ken Satoh: "Personalized Information Retrieval Systems in Legal Texts," in Proceedings of First International Workshop on SCIentific DOCument Analysis (SCIDOCA), 2016.

[9] Truong-Son Nguyen, Viet-Anh Phan, Hai-Long Trieu, Thanh-Huy Nguyen, Ngoc-Phuong Chau, Trung-Tin Phan, Le-Minh Nguyen: "Legal Information Extraction/Entailment Using SVM-Ranking and Tree-based Convolutional Neural Network", Tenth International Workshop on Juris-informatics (JURISIN), pages 177–185, 2016.

[10] Truong-Son Nguyen, Le-Minh Nguyen, Bao-Quoc Ho, and Akira Shimazu: "Recognizing logical parts in legal texts using neural architectures", In Proceedings of Eighth International Conference on Knowledge and Systems Engineering (KSE), pages 252–257, 2016.

[11] Truong-Son Nguyen, Le-Minh Nguyen: "SDP-JAIST: A Shallow Discourse Parsing system @ CoNLL 2016 Shared ", In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 143–149, 2016.

[12] Truong-Son Nguyen, Le-Minh Nguyen: "JAIST: A two-phase machine learning approach for identifying discourse relations in newswired texts", In Proceedings of the Nineteenth Conference on Computational Natural Language Learning: Shared Task (CONLL), pages 66–70, 2015.

[13] Truong-Son Nguyen, Thi-Phuong-Duyen Nguyen, Bao-Quoc Ho, Le-Minh Nguyen: "Recognizing logical parts in Vietnamese Legal Texts using Conditional Random Fields", In Proceedings of the 11th IEEE-RIVF International Conference on Computing and Communication Technologies, pages 1–6, 2015.

Awards

- The best system of Information Retrieval Task in COLIEE 2017 Live competition: Competition on Legal Information Extraction/Entailment, London, UK, June 2017.