Legal Knowledge and Information Systems
M. Araszkiewicz and V. Rodríguez-Doncel (Eds.)

doi:10.3233/FAIA190314

Neural Network Based Rhetorical Status Classification for Japanese Judgment Documents

Hiroaki YAMADA [a], Simone TEUFEL [a,b] and Takenobu TOKUNAGA [a]
[a] *School of Computing, Tokyo Institute of Technology, Japan*
[b] *University of Cambridge, Computer laboratory, U.K.*

Abstract. We address the legal text understanding task, and in particular we treat Japanese judgment documents in civil law. Rhetorical status classification (RSC) is the task of classifying sentences according to the rhetorical functions they fulfil; it is an important preprocessing step for our overall goal of legal summarisation. We present several improvements over our previous RSC classifier, which was based on CRF. The first is a BiLSTM-CRF based model which improves performance significantly over previous baselines. The BiLSTM-CRF architecture is able to additionally take the context in terms of neighbouring sentences into account. The second improvement is the inclusion of section heading information, which resulted in the overall best classifier. Explicit structure in the text, such as headings, is an information source which is likely to be important to legal professionals during the reading phase; this makes the automatic exploitation of such information attractive. We also considerably extended the size of our annotated corpus of judgment documents.

Keywords. Japanese NLP, Legal NLP, Argument understanding, Machine learning, Sentence classification, Natural language processing, Neural network, Deep learning, Rhetorical status classification

1. Introduction

Like in all other areas of life, information overload has also become problematic in the legal domain. Legal practitioners, including lawyers and judges, need to find relevant documents for their cases, and efficiently extract case-relevant information from them. In the Japanese legal system, one of the main sources used for this task is the judgment document, an important type of legal document which is the direct output from court trials and contains the judgment, the facts and the grounds[1,2]. They are typically long and linguistically complex, so that it becomes impossible to read all relevant documents carefully. Summaries of judgment documents are a solid solution to the problem, as they would facilitate the decision which documents the legal professionals should read with full attention. Our final goal is to develop methods for automatically generating such summaries.

Our project is based on the observation that the structure of the legal argument can guide summarisation. In the Japanese judgment documents, a common structure exists (Figure 1), which centres around the so-called “Issue Topic,” a legal concept corresponding to pre-defined main points which are to be discussed in a particular court case. An example for a legal case about a damage compensation case of a traffic accident in a

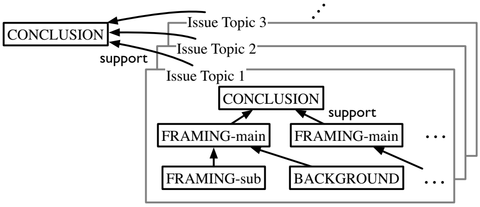

Figure 1. Argument structure of judgment document

bus travel is the question of the degree of plaintiff's own negligence. A case consists of several Issue Topics (three in the figure), and each is associated with a conclusion by the judge, and with supporting arguments for the decision. The task of argument structure extraction can be divided into four subtasks [3]: 1. *Issue Topic Identification*: find sentences that describe an Issue Topic; 2. *Rhetorical Status Classification*: determine the rhetorical status of each sentence; 3. *Issue Topic Linking*: associate each sentence with exactly one Issue Topic; 4. *FRAMING Linking*: link two sentences if one provides argumentative support for the other.

In this paper, we focus on Rhetorical Status Classification (RSC), the task of classifying sentences according to their rhetorical role (e.g. BACKGROUND or CONCLUSION). In the legal domain, this task is often seen as a preprocessing step for later tasks such as legal information extraction, extractive summarisation and argument mining [4,5,6]. We define seven RSC categories as follows; Table 1 lists them and gives an example for each. FACT covers descriptions of the facts giving rise to the case; BACKGROUND is reserved for quotations or citations of law materials (legislation and relevant precedent cases); CONCLUSION marks the decisions of the judge; and IDENTIFYING is a category used for text that states discussion topics. The primary argumentative material is contained in the two categories FRAMING-main and FRAMING-sub. FRAMING-main marks material which directly supports the judge's conclusion, whereas FRAMING-sub is one of the two categories which can support FRAMING-main (the other being BACKGROUND). These categories are crucial for downstream argumentative structure mining (task 4). Material that can not be classified into any of the above classes is covered by the OTHER category.

In our previous work, RSC performance was acceptable overall, but differed across category: in particular, in some of the most important categories for downstream tasks, performance was low. BACKGROUND, which is an important category listing relevant law materials, achieved only F=0.32, and CONCLUSION, which describes the most important argumentative sentences, F=0.39. We were also not fully satisfied with the performance of the two FRAMING categories.

In this paper, we present our improved RSC classifier for Japanese judgement documents, which uses a neural network-based architecture. One of the new information sources for our model is information coming from headings in the text. Our method is motivated by human readers' scanning behaviour during reading. We also present our new, considerably larger annotated corpus of Japanese judgements.

2. Data and Annotation

The corpus we used in previous work [3] consists of 89 Japanese judgment documents of Civil law cases from lower court cases with annotations of argumentative structure.

Table 1. Examples for RSC categories

Label	**Example (translated)**
IDENTIFYING	*Based on the agreed facts and the gist of the whole argument, we discuss each issue in the following.*
CONCLUSION	*Therefore, the plaintiff's claim is unreasonable since we just found that the officer was not negligent.*
FACT	*The duties of an execution officer are… and officer D properly conducted…*
BACKGROUND	*It is reasonable to find the officer negligent when the officer did not the appropriate… (1997/7/15 ruling of the third Petty Bench of the Supreme Court).*
FRAMING-main	*The measures performed by the officer comply with the normal procedure for inspection.*
FRAMING-sub	*It is considered that officer D entered the estate to confirm the circumstance…*

Table 2. RSC class distribution of our corpora in percent

	FACT	FR-main	FR-sub	CONC	IDEN	BACK	OTH	sent.
Previous (89 doc)	23.1	19.5	11.5	3.9	2.1	0.3	39.7	37,371
New (t&t, 110 doc)	23.5	19.1	10.6	3.8	2.0	0.3	40.6	44,677

They were sourced from website maintained by the supreme court of Japan[1] by a random selection process. Our new corpus extends this set to 120 documents (48,370 sentences, 3.2 million characters) following the same principles, and the same expert annotator (a PhD candidate in a graduate school of Japanese Law, who was paid for this work) was used. The annotation is kept consistent with the preceding paper, i.e, annotations for all four subtasks above are obtained at the same time. Category assignment is exclusive, i.e., only one category can be assigned to each sentence. We reserved ten documents out of 120 documents as development data for hyperparameter tuning. The remaining 110 documents are used for the experiments reported here. Table 2 shows the category distribution and total for our test and training corpus of 110 documents, against the previously used test and training corpus of 89 documents.

3. Conditional Random Field baseline model

Previous work on RSC in legal documents found that RSC is strongly affected by context in terms of other rhetorical roles[3,5]. We therefore use Conditional Random Fields (CRF) [7][2] as a strong baseline model.

As features, we use the seven features from [3]: the **bigram**, **sentence location**, **sentence length** features (calculated in characters). We also use 8 **modality** features based on Masuoka's (2007) modality expression classification, namely the modalities "truth judgment" (4 features; e.g, "*hazu da*" (can be expected to be) or "*beki da*" (should be)), "value judgment" (3 features), and "explanation". The **function expression** feature distinguishes the 199 semantic equivalence classes contained in the function expression dictionary by [10](such as "evidential" and "contradictory conjunction"); this covers 16,801 separate surface expressions. The **cue phrase** feature contains an additional 22 phrases

[1] `http://www.courts.go.jp/`

[2] We used Okazaki's (2011) implementation.

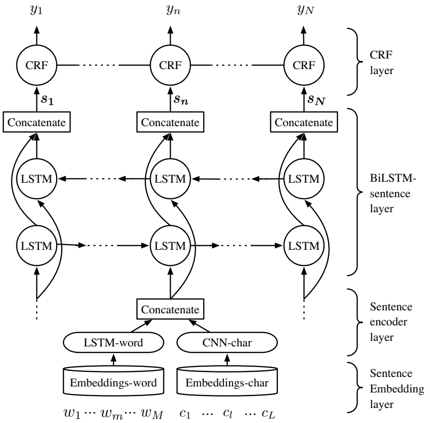

Figure 2. BiLSTM-CRF model for RSC.
$\boldsymbol{w}$: words from an input sentence;
$\boldsymbol{c}$: characters from an input sentence;
$\boldsymbol{s}$: contextualised vectors of sentences;
y: predicted RSC category.

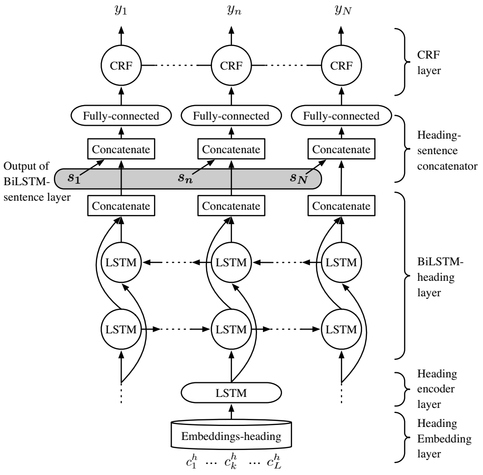

Figure 3. BiLSTM-CRF model with heading.
$\boldsymbol{c^h}$: characters from an input heading;
$\boldsymbol{s}$: contextualised vectors of sentences that are the same in Figure 2.

from a textbook used during the training of judges [11] and from five judgment documents not included in the training and test data, and the **law names** feature, which distinguishes 494 specific law names as features and adds a binary feature indicating the presence of *any* law name in the sentence. A document is then input into the CRF model as a sequence of sentences, where each sentence is represented by the features above.

4. BiLSTM-CRF based model

We tested our BiLSTM-CRF based sentence sequence labelling architecture presented here against the baseline model.

BiLSTM-CRF architectures have recently become the standard deep learning method for modelling sequences is the Bidirectional-LSTM(BiLSTM) [12], which can encode both preceding and succeeding context; they have been used for Named Entity Recognition (NER) and POS-tagging [13]. Context in the form of surrounding text as well as surrounding labels can be taken into account with this architecture: past and future input features can be modelled through BiLSTM layers, whereas sequences of labels can be modelled through a CRF layer. Variants of BiLSTM-CRF differ in how to encode the token vectors which is the input to a sequence level BiLSTM layer: [14] uses a Convolutional neural network (CNN)-based character-level representation in addition to word embeddings, whereas [15] uses a character-level representation which is encoded by another BiLSTM encoder. Our BiLSTM-CRF model has three main components, a sentence encoder layer, a BiLSTM-sentence layer, and a CRF layer (Figure 2).

Sentence encoder layer Our target units are sentences, not words as in the POS-tagging and NER task, so they need to be encoded into vectors before passing them to the BiLSTM layer. The sentence encoder layer consists of two components, LSTM-word and CNN-char. LSTM-word takes word embeddings of sentences as input and outputs the summarised vector for each sentence. CNN-char is a simple CNN with one layer of convolution [16], which takes character embeddings of sentences as input and generates the summarised vector for each sentence. In pre-experiments, using both LSTM-word and CNN-char showed improved performance over using either of these on their own. While LSTM-word should encode the overall meaning of input sentences, CNN-word should capture the characteristic combinations of characters such as typical combinations of Chinese characters and Hiragana characters. Outputs from LSTM-word and CNN-char are concatenated.

BiLSTM-sentence layer / CRF layer We use the architecture proposed in [13]. The BiLSTM-sentence layer takes a sequence of sentences as input and concatenates the hidden state vectors from two LSTMs run bidirectionally; the output of this step should correspond to a contextualised representation of the input sentence vector, which is then input to a CRF layer that computes the final output.

Dropout For regularisation, we include dropout [17] after the LSTM-word and the BiLSTM-sentence layer.

4.1. Input data and embeddings

The inputs to the sentence encoder layer are vector representation of words and characters. As for word inputs, we use the SentencePiece algorithm [18] to tokenise a sentence into tokens, a step necessitated by the fact that the Japanese script does not use an explicit word separator. SentencePiece is an unsupervised text tokeniser which allows us to tokenise without any pre-defined dictionaries. We trained the tokeniser on 15 thousands Civil and Criminal law judgment documents that are published during 1989—2017, using the same web source as our test and training corpus, but excluding the documents used in it (note that the domains differ slightly as our test and training corpus consists only of Civil Law cases). The tokenised words are then input into the embedding layer.

As for character inputs, we simply split a sentence into characters and input them to the embedding layer. The meaning-bearing part of most open-class Japanese words is due to one or more Chinese characters, which are semi-compositionally combined. The characters themselves might therefore contribute additional meaning components and similarities between words beyond the word identities themselves. Each embedding layer converts the input to embedding vectors, which form the input to the sentence encoder. We initialise the embedding layer for characters with GloVe [19] vectors pre-trained with judgment documents of Civil law cases published in the last 14 years (2004–2017).[3]

4.2. Input and output handling

An input to the model is a sequence of sentences. We restrict the length of the sequence to an odd number w.[4] We obtain a sequence of inputs by sliding the size w window from the beginning to the end of the document, sentence by sentence. The n-th input from document D can be represented as $Q_D^n(w) = \{S_D^{n-(w-1)/2}, ..., S_D^n, ..., S_D^{n+(w-1)/2}\}$, where S_D^i is the i-th sentence in document D. At the beginning and the end of the document,

[3] We found in pre-experiments that halving the corpus we used for the tokenisation experiment (disregarding the older half) lead to better results. The target embedding vector dimension is set to 300.

[4] Preliminary experiments where an entire document was input as a single sequence showed low results. The average length of documents was 403.1 lines, which proved too long even for LSTMs with their ability to store a good amount of long-term context.

we fill padding tokens if necessary. We have w predictions in an output for each sentence according to its relative position in the input. We use the prediction that is located in the middle of output.[5]

5. BiLSTM-CRF based model with Headings

We next present a new model which uses the information contained in the documents' headings. Exploiting explicit structural information from the text, such as headings, could model the reading strategy of legal professionals. In particular, we hypothesise that when a human reader notices a new heading in a document, they might interpret this as a signal of rhetorical status change.

In addition to the components of the BiLSTM-CRF model, a dedicated network for handling heading information is added to the model (Figure 3). The network consists of three parts, heading encoder, BiLSTM-heading, heading-sentence concatenator. The heading encoder is a character-based LSTM encoder which summarises the input character embeddings of a heading and outputs a heading vector. The Heading BiLSTM is similar to the BiLSTM-sentence layer, which generates a contextualised representation of headings per input, but is activated only for headings. It does so by inputting the sentence itself to the the upper network layers; otherwise, a special character "_" is used, which signals the non-existence of a heading. The outputs from the BiLSTM-sentence layer and the heading BiLSTM are concatenated and input to a fully-connected layer. The CRF layer then receives the output from the fully-connected layer.

As headings are not explicitly annotated in our corpus, we detect them automatically using a binary rule-based heading detector based on the presence of sentence-final punctuation and sentence length. The detector's performance of finding headings was $F = 0.89$ ($R = 0.99$, $P = 0.81$), measured on all 2,061 lines in 5 random documents (manually annotated by the first author). 622 lines were headings (lines which only contain headings and nothing else) and 1,439 non-headings (either normal sentences, or lines which erroneously contain both a heading and the beginning of a normal sentence).

6. Experiment

6.1. Experimental setting

We use 110 documents from our corpus for training and testing of our two BiLSTM models described in section 4 and 5. Hyperparameters of BiLSTM-CRF models are empirically tuned using the development data (10 documents). The hyperparameters we use for the experiment are shown in Table 3. We use five-fold cross-validation at the document level.

In order to make sure that any performance improvement over our previous work is not only due to the use of heading information per se but to the architecture, we also make the heading information available to the CRF, in the form of a binary feature expressing heading existence, a variant we call [6]. This means that we report results for a total of four models (**CRF, CRF+H, BiLSTM, BiLSTM+H**). We test significance of macro-

[5] Due to a quirk in the experiments, we only pad at the beginning of documents, not at the end. This leads to some cases in each document where the predicted item is not in the middle of the outputs. In those cases, we use the last prediction of the output.

[6] CRF+H also gets the strings of the headings through bigram feature.

Table 3. Hyperparameters for BiLSTM-CRF models

Hyperparameters	values
epochs	1
word emb dim	300
char emb dim	300
LSTM-word	64
CNN-char window	5
CNN-char channels	256
LSTM-word dropout	0.2
BiLSTM-sent	128 + 128
BiLSTM-sent dropout	0.2
heading emb dim*	64
heading encoder*	64
BiLSTM-heading*	64 + 64
final cocat*	128

∗ If applicable

averaged F measure using a Monte Carlo paired permutation test randomisation at the sentence level with R=100,000 samples at a significance level of $\alpha = 0.05$ (two-tailed).

6.2. Results

Overall results are shown in Table 4. BiLSTM-CRF+H (F=0.654 with setting $w = 11$) significantly outperforms both CRF (F=0.630) and CRF+H (F=0.632), showing that the Deep Learning architecture with heading information indeed represents an overall improvement. This effect holds also without heading information: BiLSTM-CRF ($w = 21$) (F=0.651) is significantly better than CRF (F=0.630) and CRF+H (F=0.632). The BiLSTM-CRF model family overall outperforms the CRF model family.

Although the macro-averaged F performance difference between BiLSTM and BiLSTM-CRF+H is not significant, several individual categories show significant improvement when heading information is added (see Table 5), namely BACKGROUND (F=0.341), FRAMING-main (F=0.651) and CONCLUSION (F=0.449). These are three of the four categories we care about most, as they carry most information for the legal argumentation and form a basis of our further planned processing in this application.

However, this success is paid for with a significant decrease (from F=0.527 to 0.474) for the FRAMING-sub category. These results notwithstanding, we still promote the heading-enabled BiLSTM as our preferred model, as the three improved categories also include the previously weakest of those 4 categories (BACKGROUND increased from F=0.319 to 0.341). With a roughly equal performance in CONCLUSION and FRAMING-sub of both over F=0.45, this leaves us overall in a better situation than without the heading information.

The confusability between FRAMING-main and FRAMING-sub should be one of the main reasons for the remaining errors. Table 6 shows the confusion matrix of the BiLSTM-CRF+H. 1,990 out of 4,727 FRAMING-sub sentences (42.0%) are wrongly classified as FRAMING-main. According to the agreement study of RSC annotation scheme from the previous study [3], the distinction between those two categories is hard even for human annotators. The problem is that the categories both appear in similar locations and have similar surface characteristics e.g. "therefore" phrase in Japanese.

7. Related Work

Rhetorical Status Classification is a commonly used approach in legal text processing for associating text pieces with their rhetorical status. Our rhetorical annotation scheme of six categories plus the OTHER category is an adaptation of previous schemes for the UK law system [4] and Indian law system [5]. For the automatisation of RSC, CRF and other machine learning models have been employed. For RSC of the UK law system, Hachey and Grover used various supervised machine learning systems, achieving the best

Table 4. Macro-averaged results for models

Models	Precision	Recall	F
CRF	0.681	0.603	0.630
CRF + Heading	0.685	0.605	0.632
BiLSTM-CRF ($w = 11$)	0.663	0.635	0.647
BiLSTM-CRF ($w = 21$)	**0.686**	0.629	0.651
BiLSTM-CRF ($w = 31$)	0.673	0.615	0.638
BiLSTM-CRF + Heading ($w = 11$)	0.679	**0.636**	**0.654**
BiLSTM-CRF + Heading ($w = 21$)	0.657	0.628	0.640
BiLSTM-CRF + Heading ($w = 31$)	0.653	0.620	0.633

Table 5. Results of models by classes (F)

Category	CRF	BiLSTM-CRF	BiLSTM-CRF+H
BACKGROUND	**0.344**	0.319	0.341
CONCLUSION	0.381	0.415	**0.449**
FACT	0.853	**0.890**	0.879
FRAMING-main	0.594	0.642	**0.651**
FRAMING-sub	0.471	**0.527**	0.474
IDENTIFYING	0.792	0.798	**0.806**
OTHER	0.972	0.969	**0.975**

BiLSTM-CRF is $w = 21$ and BiLSTM-CRF+H is $w = 11$.

Table 6. Confusion Matrix of BiLSTM-CRF + Heading ($w = 11$)

		Prediction							
		BGD	CCL	FCT	FRm	FRs	IDT	OTR	Total
Gold	BGD	38	0	19	38	29	0	3	127
	CCL	0	699	42	847	28	6	90	1,712
	FCT	3	36	9,544	500	181	15	235	10,514
	FRm	18	548	745	5,836	1,214	44	132	8,537
	FRs	33	31	628	1,990	1,944	49	52	4,727
	IDT	2	15	30	96	53	710	24	930
	OTR	2	73	191	76	18	7	17,763	18,130
	Total	96	1,402	11,199	9,383	3,467	831	18,299	44,977

results with C4.5 [20] with only the location feature (F=0.65); the second-best (F=0.61) was achieved using a Support Vector Machine [21] with all features (location, thematic words, sentence length, quotation, entities and cue phrases). As for RSC of Indian law system, a CRF classifier with various features similar to our CRF model achieved F=0.82.

Walker et al develop a rule-based RSC classifier from a small amount of labelled data [6]. Their task is to identify rhetorical roles of sentences such as “Finding”, which states whether a propositional condition of a legal rule is determined to be true, false or undecided, “Evidence”, such as the testimony of a lay witness or a medical record, “Reasoning” which reports reasoning parts underlying the findings of fact (i.e. a premise), “Legal-Rule” which states legal rules, and “Citation” which references legal authorities or other law materials, and “Others”. There are close similarities to our categories. 530 sentences were used to develop a rule set set for their classifier, and the paper reports the comparison between their low-cost rule-based classifier (F=0.52).

Some F-measures from previous studies are higher than ours, but this mirrors the difficulty of our task. None of the other schemes makes such fine distinctions as we do, particularly in the lower levels of argumentative support such as those expressed by the FRAMING-main vs FRAMING-sub distinction.

Another piece of work performs *deontic* sentence classification in contract documents [22], using a hierarchical RNN-based architecture. The sentences are classified into “Obligation”, “Prohibition”, “Obligation List Intro”, “Obligation List Item”, and “Prohibition List Item”. The model is based on a BiLSTM-based sequential sentence classifier which considers both the sequence of words in each sentence and the sequence of sentences like our models, but it does not employ a label sequence optimiser such as our CRF layer.

Outside the legal document processing community, RSC is often used in the area of scientific paper processing for the extraction of relevant material and for summarisation. An RNN-based model similar to ours has been proposed for the RSC of sentences in medical scientific abstracts [23]. Our model shares the basic design (a sentence encoder, a context encoder, and a CRF layer) with this model; however, their model does not consider heading information.

8. Conclusion

In this paper, we proposed to apply a BiLSTM-CRF based model for rhetorical status classification. It performs RSC with sequential labelling by taking inter-sentence level context into account. We also proposed to add a dedicated network which conveys contextualised heading information, after headings have been recognised by a simple automatic heading detector. The model showed significant improvements from the plain BiLSTM-CRF model in BACKGROUND, FRAMING-main and CONCLUSION. We also extended the size of our annotated corpus of Japanese judgment documents. The resulting system showed a significant improvement from our CRF based baseline models.

There are several possible directions for future work. One of these is to train our model with curriculum learning strategy [24]. Curriculum learning is a training approach that exposes a model by giving training examples in a meaningful order, gradually increasing difficulty. RSC seems to fit this training scheme very well, as it shows various patterns of sequences from simple ones such as category repetitions (“FACT, FACT, FACT . . . ”) to more complicated ones such as “FRAMING-sub, FRAMING-sub, FRAMING-main, FRAMING-sub, BACKGROUND . . . ”. Curriculum learning might therefore help our model to learn how to distinguish difficult categories (e.g. FRAMING-sub v.s. FRAMING-main) in an efficient way. Also, we plan to conduct an extrinsic evaluation with a summarisation task by lawyers, which uses the results of the RSC.

Acknowledgments. This work was supported by Tokyo Tech World Research Hub Initiative (WRHI) Program of Institute of Innovative Research, Tokyo Institute of Technology.

References

[1] Ministry of Justice, Japan, “Form of Rendition”, Code of Civil Procedure, Article 252.

[2] Ministry of Justice, Japan, “Judgment Document”, Code of Civil Procedure, Article 253.

[3] H. Yamada, S. Teufel and T. Tokunaga, Building a corpus of legal argumentation in Japanese judgement documents: towards structure-based summarisation, *Artificial Intelligence and Law* **27**(2) (2019), 141–170.

[4] B. Hachey and C. Grover, Extractive summarisation of legal texts, *Artificial Intelligence and Law* **14**(4) (2006), 305–345.

[5] M. Saravanan and B. Ravindran, Identification of Rhetorical Roles for Segmentation and Summarization of a Legal Judgment, *Artificial Intelligence and Law* **18**(1) (2010), 45–76.

[6] V.R. Walker, K. Pillaipakkamnatt, A.M. Davidson, M. Linares and D.J. Pesce, Automatic Classification of Rhetorical Roles for Sentences: Comparing Rule-Based Scripts with Machine Learning (2019).

[7] J.D. Lafferty, A. McCallum and F.C.N. Pereira, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in: *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 282–289. ISBN ISBN 1-55860-778-1.

[8] N. Okazaki, CRFsuite: a fast implementation of Conditional Random Fields (CRFs), 2007.

[9] T. Masuoka, *Nihongo Modariti Tankyu (Japanese Modality Investigations)*, Kuroshio shuppan, 2007.

[10] S. Matsuyoshi, S. Sati and T. Utsuro, A Dictionary of Japanese Functional Expressions with Hierarchical Organization, *Journal of Natural Language Processing* **14**(5) (2007), 123–146.

[11] Judicial Research and Training Institute of Japan, *The guide to write civil judgements (in Japanese)*, 10th edn, Housou-kai, 2006.

[12] A. Graves and J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, *Neural networks* **18**(5–6) (2005), 602–610.

[13] Z. Huang, W. Xu and K. Yu, Bidirectional LSTM-CRF Models for Sequence Tagging, *CoRR* **abs/1508.01991** (2015).

[14] X. Ma and E. Hovy, End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2016, pp. 1064–1074.

[15] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, Neural Architectures for Named Entity Recognition, in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, San Diego, California, 2016, pp. 260–270.

[16] Y. Kim, Convolutional Neural Networks for Sentence Classification, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research* **15**(1) (2014), 1929–1958.

[18] T. Kudo and J. Richardson, SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 66–71.

[19] 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)*, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543.

[20] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN ISBN 1-55860-238-0.

[21] C. Cortes and V. Vapnik, Support-vector networks, *Machine learning* **20**(3) (1995), 273–297.

[22] I. Chalkidis, I. Androutsopoulos and A. Michos, Obligation and Prohibition Extraction Using Hierarchical RNNs, in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 254–259.

[23] D. Jin and P. Szolovits, Hierarchical Neural Networks for Sequential Sentence Classification in Medical Scientific Abstracts, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3100–3109.

[24] Y. Bengio, J. Louradour, R. Collobert and J. Weston, Curriculum Learning, in: *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, ACM, New York, NY, USA, 2009, pp. 41–48. ISBN ISBN 978-1-60558-516-1.